

FIG. 1A

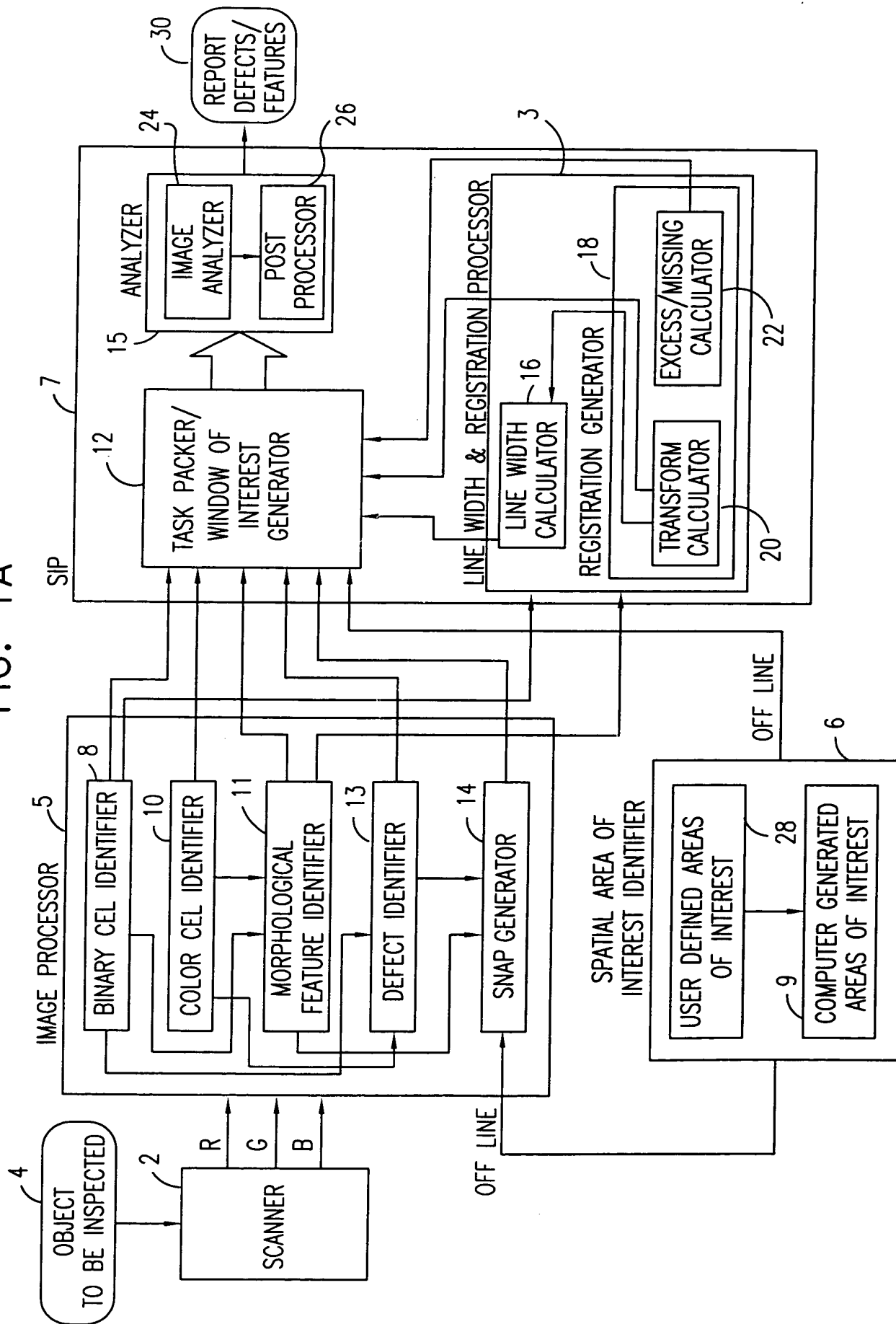


FIG. 1B

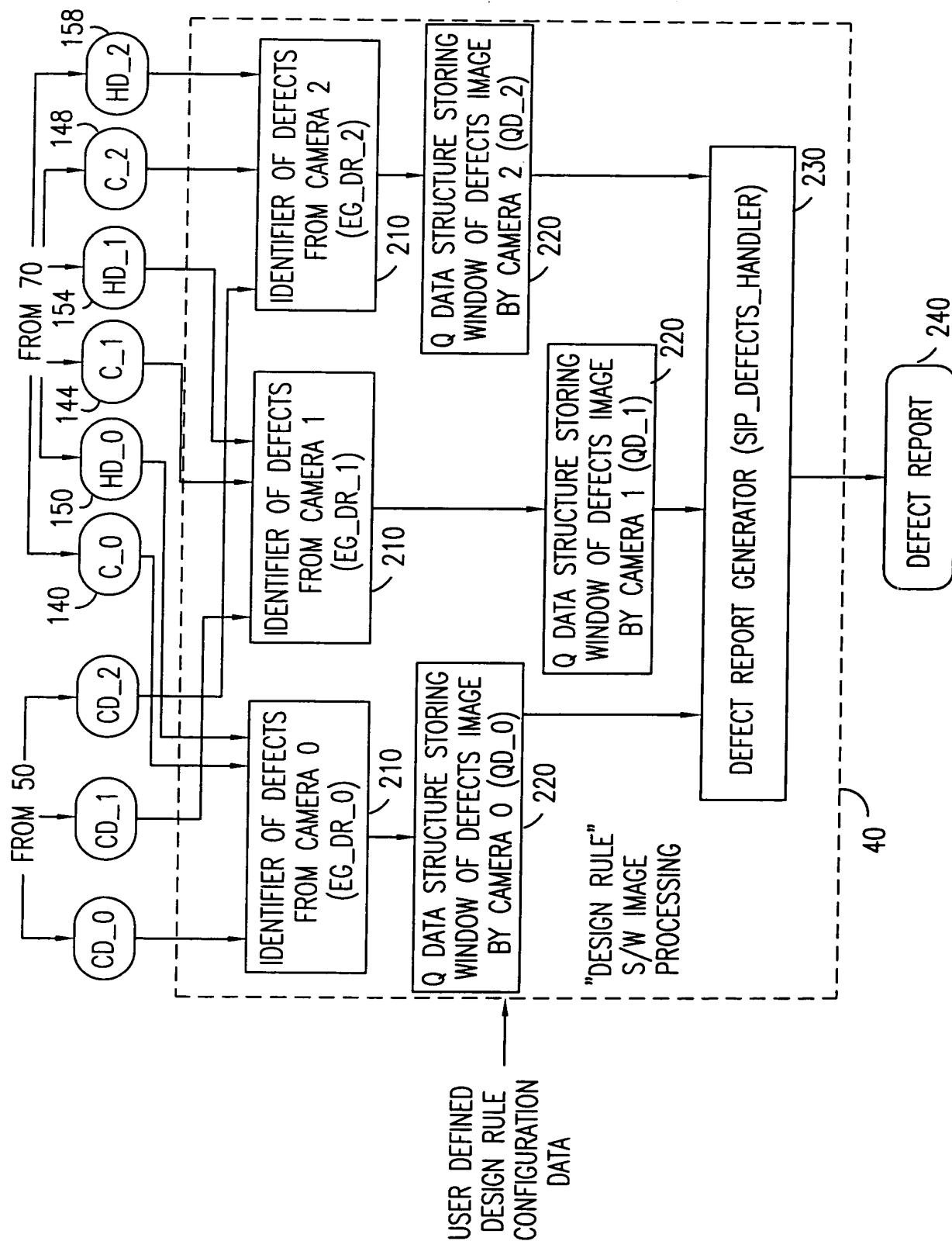
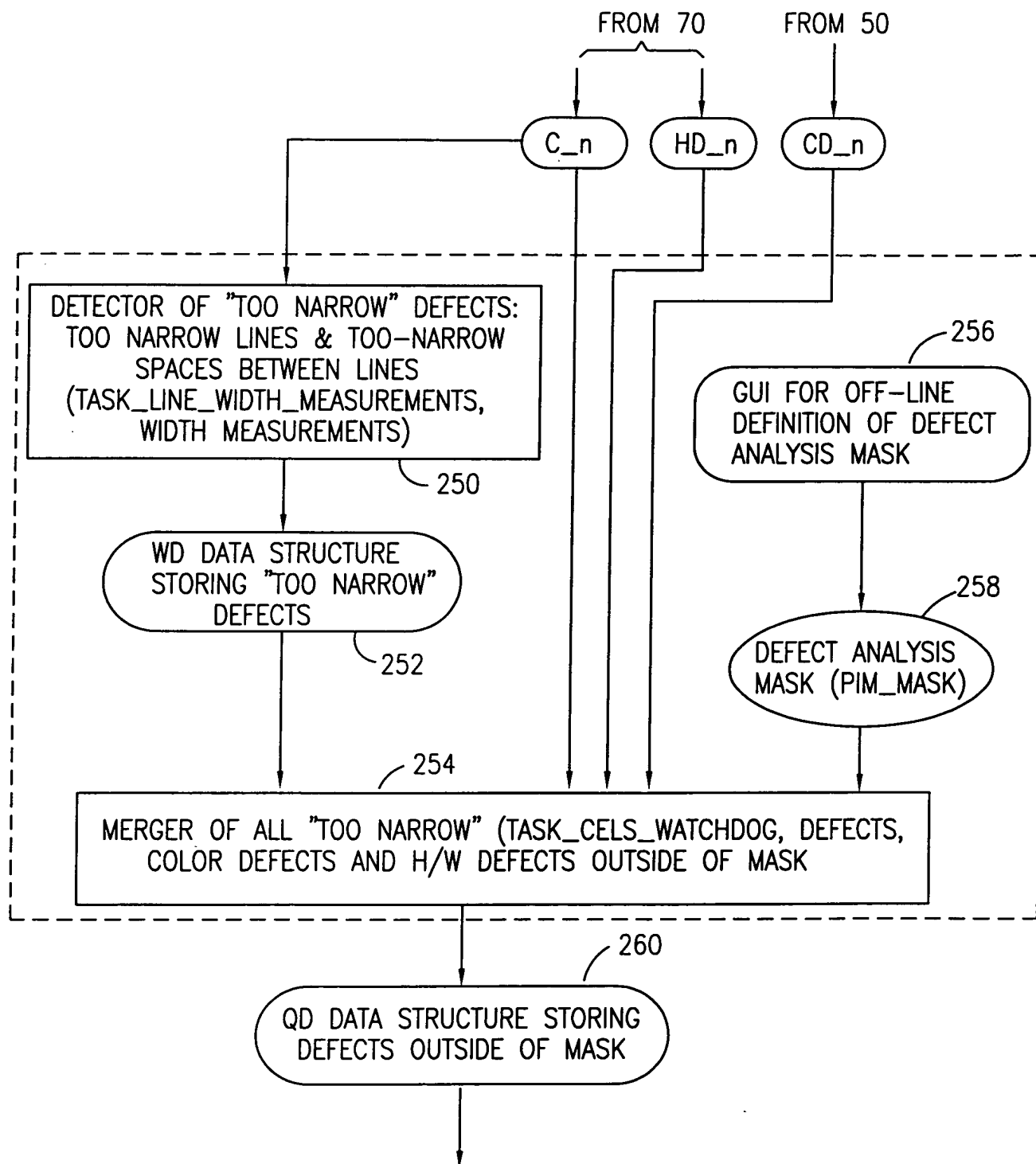
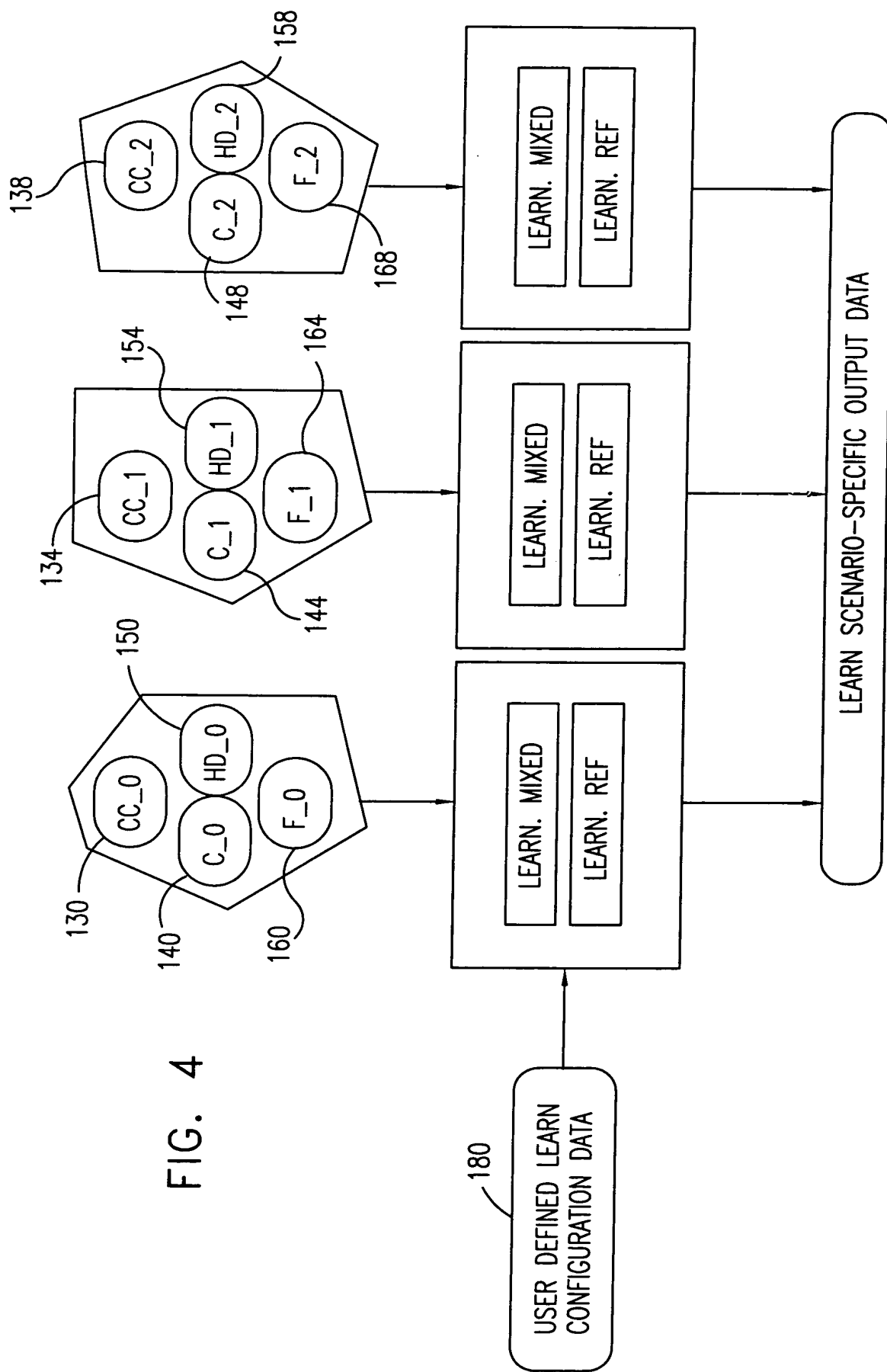


FIG. 2

FIG. 3





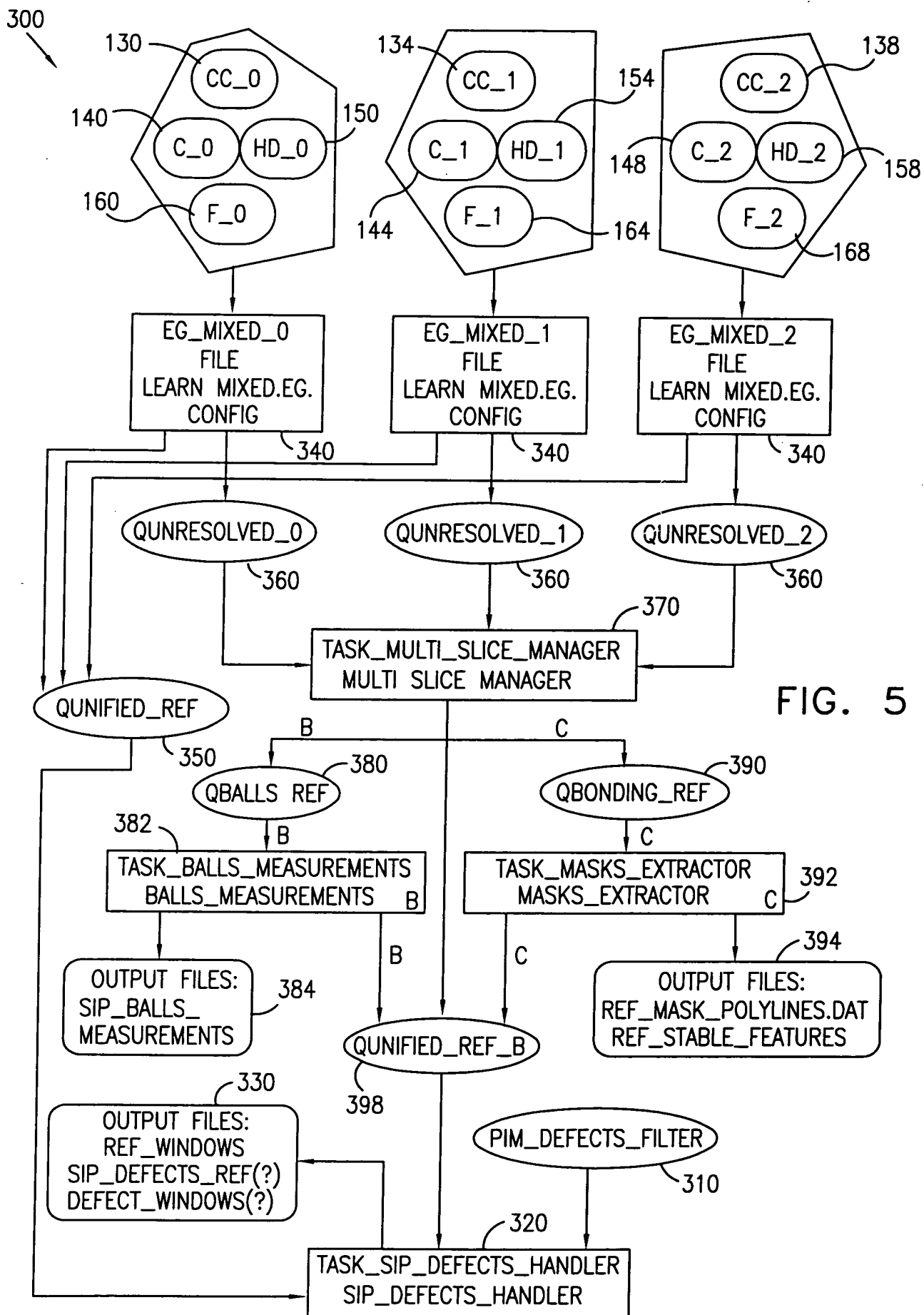
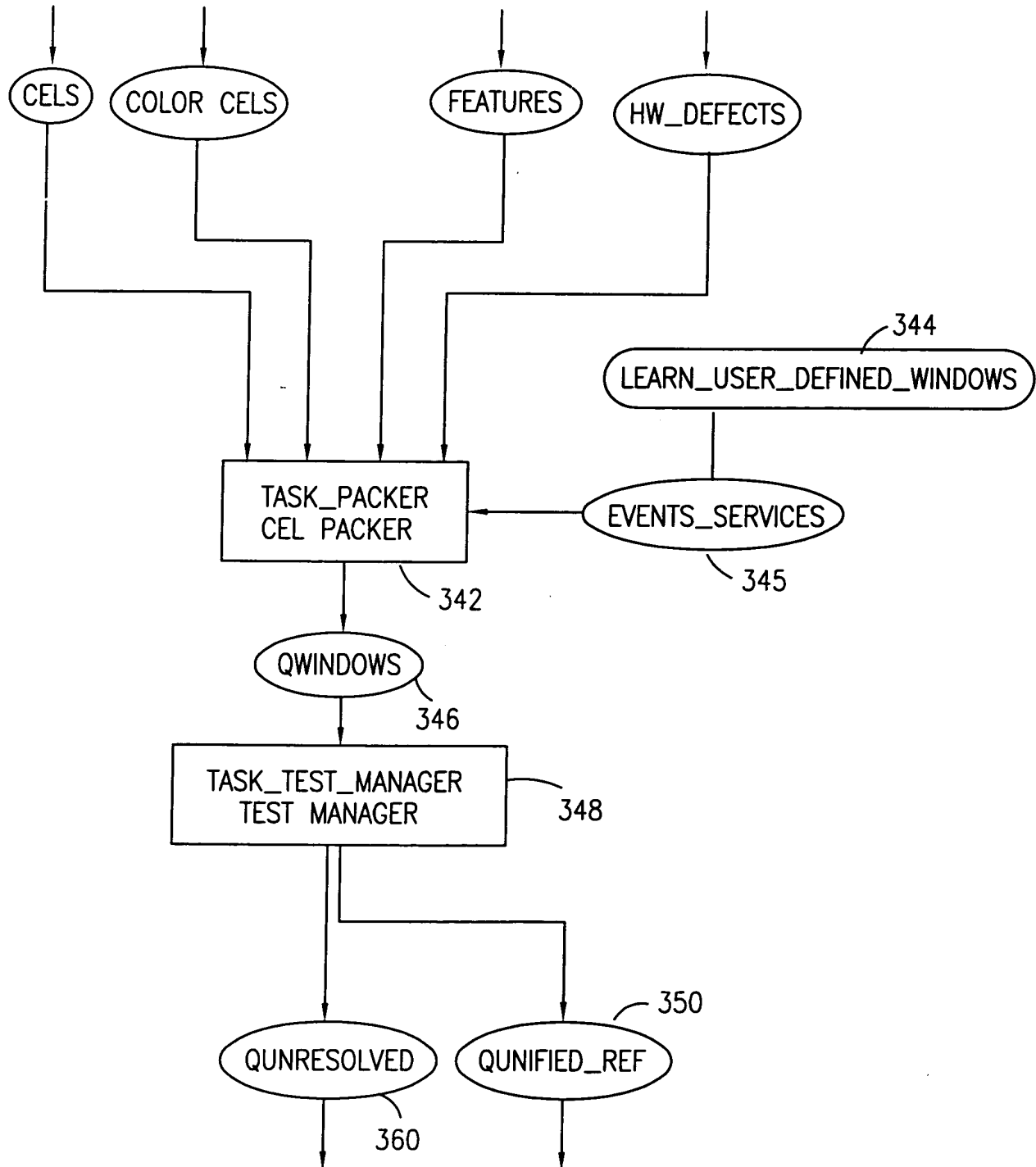


FIG. 5

FIG. 6



[illegible]

Window type	Functions attached	Remarks
target	<u>target lrn:</u> connect_closed (Func_connected_components<CEL>) vectorize (Func_angle_vectorizer<Cel>) target_reference (currently does nothing) trans2ref (Func_trans2ref) forward2target_analyze (Func_forward)	Produces connected components of raw CELs. Vectorizes them into polylines. Then transforms vectorized polylines into reference aligned coordinated system (removing all non transformable data). Finally the window is forwarded into destination unresolved and the function that is attached to the window at its new destination is target_analyze.
bonding_area	<u>ii lrn</u> connect_closed (Func_connected_components<CEL>) vectorize (Func_angle_vectorizer<CEL>) connect_open_color (only in enable_color_masking mode) (Func_connected_components<Color_cel>) vectorize_color (only in enable_color_masking mode) (Func_angle_vectorizer_<Color_cel>) stp_reference trans2ref (Func_trans2ref) forward2bonding_analyze (Func_forward)	Produces connected components of raw CELs, vectorizes them into polylines. If in enable_color_masking mode is then connected components of raw color CELs are also computed, and then vectorize them into polylines. The function stp_reference of type Func_stp_top_down_ref is called to learn bonding area and to create the single camera component of the reference including bonding pads windows, masks zones and stable features for registration. Then all transformable data is transformed into reference aligned coordinate system (removing all non transformable data). Finally the window is forwarded into destination unresolved and the function that is attached to the window at its new destination is bonding_analyze.
chip_area	nop (nop function)	Nothing is done with this window. It is only used within bonding_area window in function stp_reference.

$$\begin{array}{ccccccc} \{f^{(1)}_1\} & \{f^{(2)}_1\} & \{f^{(3)}_1\} & \{f^{(4)}_1\} & \{f^{(5)}_1\} & \{f^{(6)}_1\} & \{f^{(7)}_1\} \\ \{f^{(1)}_2\} & \{f^{(2)}_2\} & \{f^{(3)}_2\} & \{f^{(4)}_2\} & \{f^{(5)}_2\} & \{f^{(6)}_2\} & \{f^{(7)}_2\} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \{f^{(1)}_n\} & \{f^{(2)}_n\} & \{f^{(3)}_n\} & \{f^{(4)}_n\} & \{f^{(5)}_n\} & \{f^{(6)}_n\} & \{f^{(7)}_n\} \end{array} \quad \text{in} \quad \{f^{(1)}_1\} \cup \{f^{(2)}_1\} \cup \{f^{(3)}_1\} \cup \{f^{(4)}_1\} \cup \{f^{(5)}_1\} \cup \{f^{(6)}_1\} \cup \{f^{(7)}_1\}$$

Window type	Functions attached	Remarks
balls	<u>balls ins</u> connect_open (Func_connected_components<CEL>) vectorize (is needed only for debugging and visualization of balls algorithms) Func_angle_vectorizer<CEL>) circles_process (Func_circles_process) trans2ref (Func_trans2ref) forward2balls_analyze (Func_forward)	Produces connected components of raw CELs. Then vectorize them into polylines. This step is not really needed for the algorithm but it provides a nice picture to look at when visualizing the outcome of the algorithm. The function circles_process of type Func_circles_process is called to model the connected components as what are called generalized circles (balls). Then all transformable data is transformed into reference aligned coordinate system (removing all non transformable data). Finally the window is forwarded into destination unresolved and the function that is attached to the window at its new destination is balls_analyze.
cavity	<u>alarms lrn</u> watchdog-func (Func_watchdog) trans2ref (Func_trans2ref) forward2ref_if_in_camera (Func_forward)	Function watchdog checks to see if there are CELs inside the window. Any CEL found inside the window is reported as defect. Then all transformable data is transformed into reference aligned coordinate system (removing all non transformable data). Finally, if the window is completely within camera boundary then the window is forwarded into destination reference and the function that is attached to the window at its new destination is nop. If the window is not within camera boundary then the window is not forwarded.

FIG. 8

Function name	Functions executed	Remarks
balls_analyze	<p>analyze_circles</p> <p>balls_create_ref</p> <p>balls_display (for display and debug only)</p> <p>merge_defects</p> <p>forward2balls_ref (Func_forward)</p>	<p>Classifies circles based on data coming from three cameras. Creates circles reference. Merges all defects from cameras into main unified data.</p> <p>Finally the window is forwarded into destination unified_balls_reference_defects_queue and the function that is attached to the window at its new destination is strip_balls.</p>
bonding_analyze	<p>analyze-bonding-side</p> <p>forward2bonding_ref (Func_forward)</p>	<p>Megre data from all three cameras into unified reference format.</p> <p>Finally the function is forwarded into destination unified_bonding_reference_defects_queue and the function that is attached to the window at its new destination is strip-bonding.</p>
target_analyze	<p>analyze_target</p> <p>forward2target_ref (Func_forward)</p>	<p>Megre data from all three cameras into unified reference format.</p> <p>Finally the function is forwarded into destination unified_reference_defects_queue and the function that is attached to the window at its new destination is strip_target.</p>

004000"9422200

FIG. 9

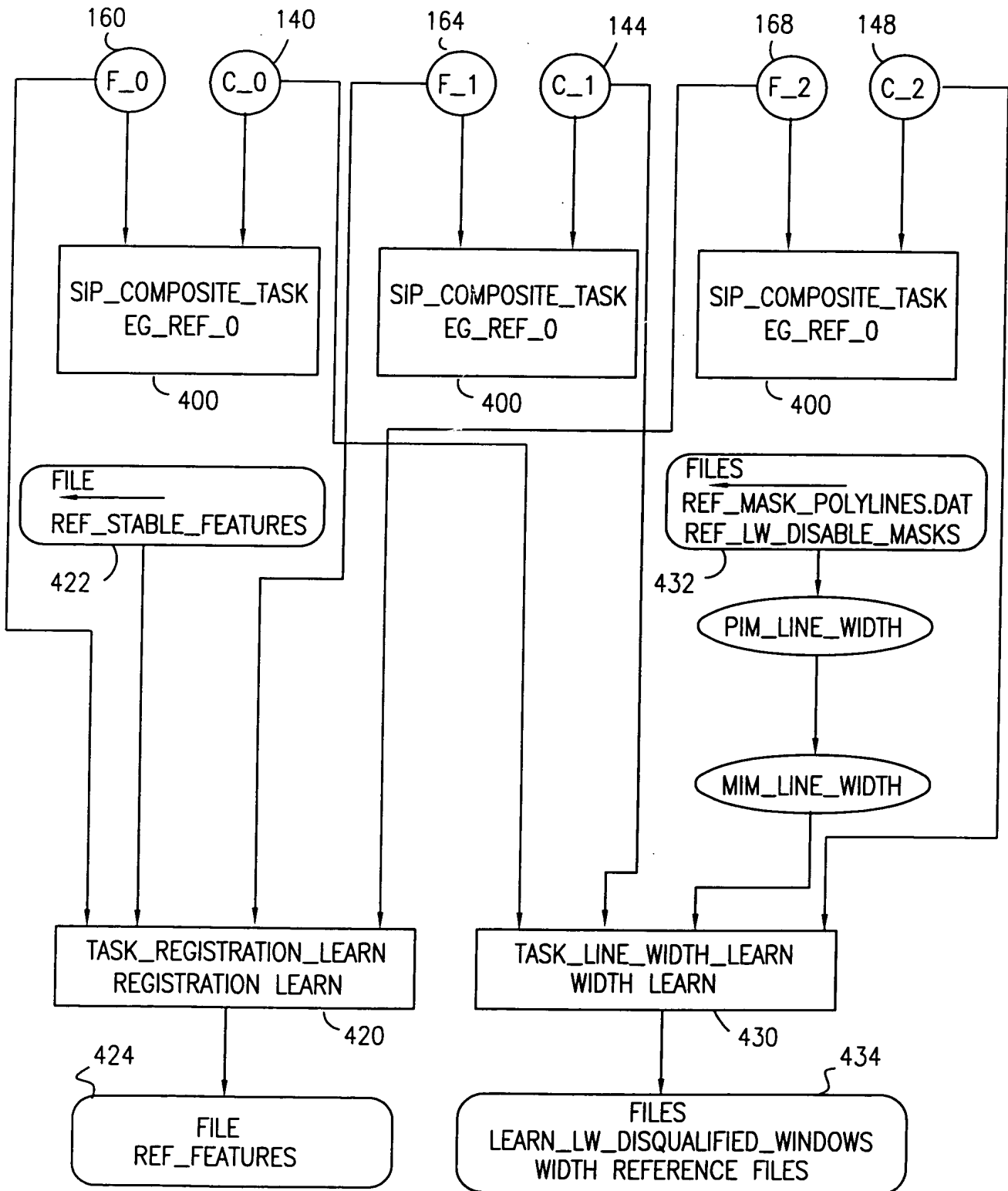


FIG. 10

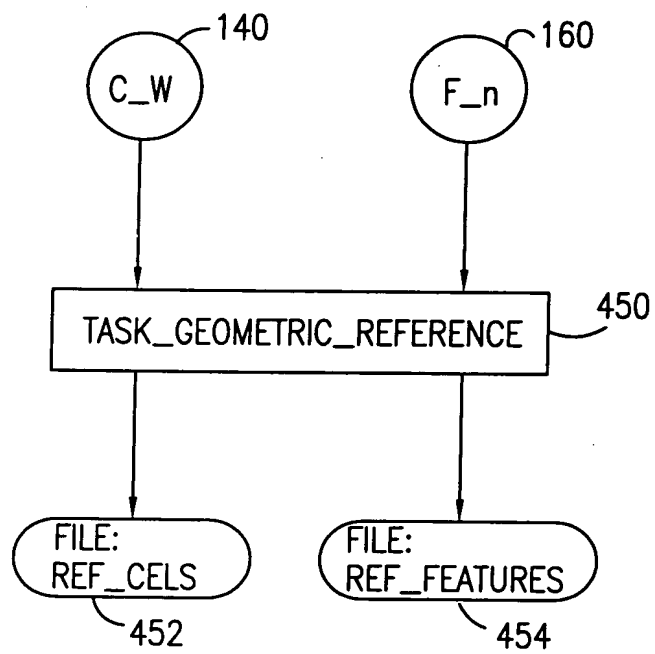


FIG. 12

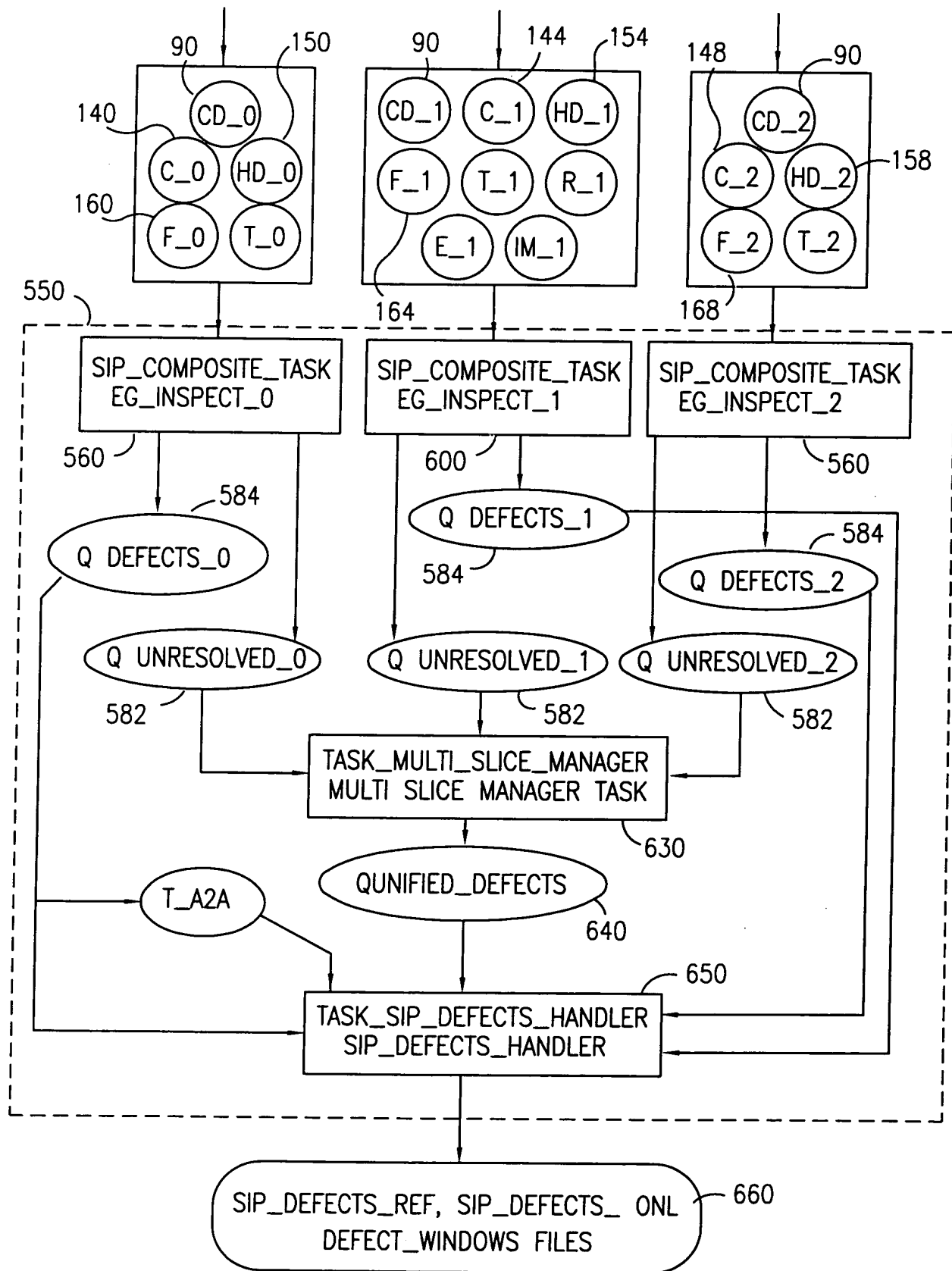


FIG. 13

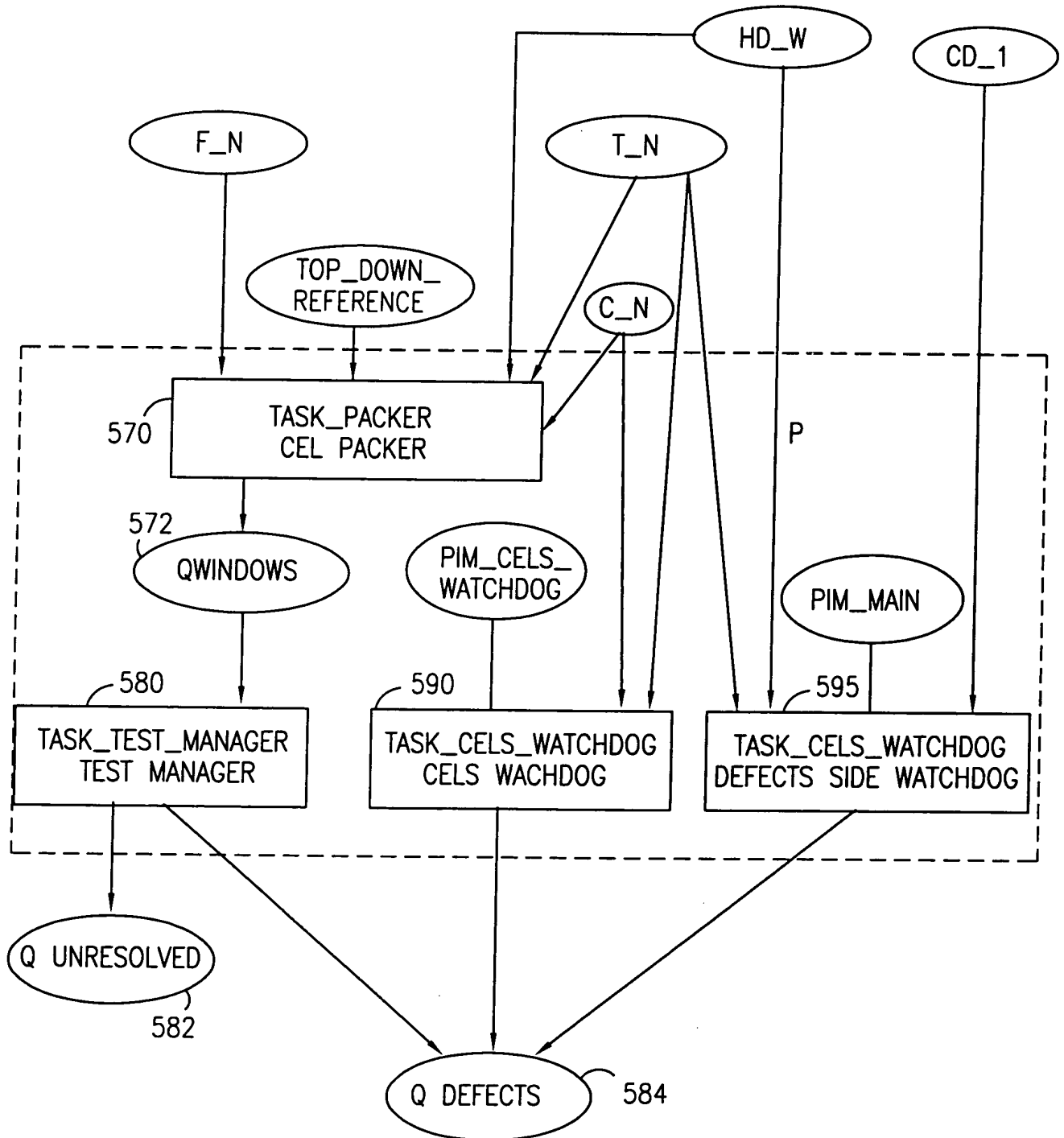
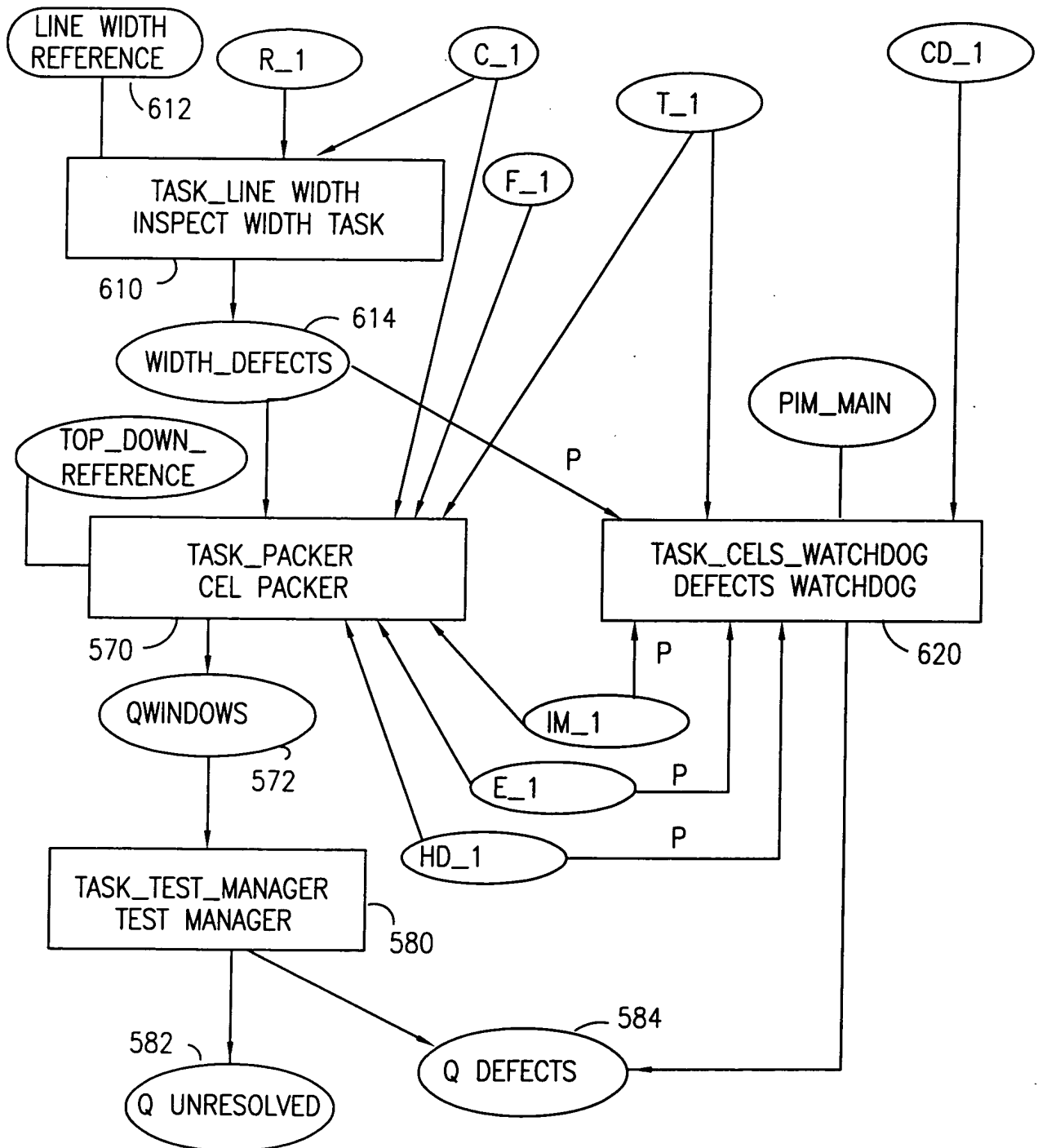


FIG. 14



1. *What is the purpose of the study?*
 2. *What are the research questions or hypotheses?*
 3. *What is the study design?*
 4. *What is the sample size and how was it selected?*
 5. *What are the variables being studied?*
 6. *What are the data collection methods?*
 7. *What are the results of the study?*
 8. *What are the conclusions and implications of the study?*

Window type	Functions attached	Remarks
target	<p><u>poly compare target:</u></p> <p>polylines_comp_aaa_target (func_polylines_comp_aaa)</p> <p>defects_handler (Func_defects_handler)</p> <p>defectsfiler (Func_defects_filer)</p> <p>forward2defects (Func_forward)</p>	<p>Function polylines_comp_aaa_target of type func_polylines_comp_aaa do a CEL2VEC comparison between reference polygons and online CELs based on application target tolerances.</p> <p>Function defects_handler of type Func_defects_handler is a post processing function that decides which of the defects reported by the various defect detectors (nick, protrusion, width defect CEL2VEC, excess/missing, etc.) are real Sip_defects. Defects found by a detector are rechecked according to zone specific application criteria.</p> <p>Function defectsfiler is of type Func_defects_filer is called to filter out any defects. It uses PIM_main and filters out all defects found in one or more regions of the following: unstable, mask_region, power_line_default.</p> <p>Finally, if any Sip_defects are inside the window, then the window is forwarded into destination defects and the function that is attached to the window at its new dimension is nop (nothing to do). If no defects are found, then the window is not forwarded at all.</p>
pad_ref	<p><u>poly compare pads</u></p> <p>polylines_comp_aaa_target (func_polylines_comp_aaa)</p> <p>defects_handler (Func_defects_handler)</p> <p>defectsfiler(Func_defects_filer)</p> <p>forward2defects(Func_forward)</p>	<p>This function connected to this window type is similar to the function connected to the target window above. The only difference is that parameter applications are used for bonding pads to control the behavior of the CEL2VEC function.</p>

FIG. 15B

Window type	Functions attached	Remarks
balls	<u>balls ins</u> connect_open (Func_connected_components<CEL>) vectorize (is needed only for debugging and visualization of balls algorithms) Func_angle_vectorizer<CEL>) circles_process (Func_circles_process) trans2ref (Func_trans2ref) forward2balls_analyze (Func_forward)	Produces connected components of raw CELs. Then they are vectorized into polylines. This step is not really needed for the algorithm but it provides a nice picture to look at when visualizing the outcome of the algorithm. The function circles_process of type Func_circles_process is called to model the connected components and are called generalized circles (balls). Then all the transformable data is transformed into reference aligned coordinate system (removing all non transformable data). Finally the window is forwarded into destination unresolved and the function that is attached to the window at its new destination is balls_analyze.
cavity	<u>watchdog</u> watchdog_func (Func_watchdog) defectsfilter (Func_defects_filter) forward2defects (Func_forward)	Function watchdog checks to see if there are CELs inside the window. Any CEL which is found inside the window is reported as a defect. Function defectsfilter is of type Func_defects_filter is called to filter out any defects. It uses PIM_main and filters out all defects found in one or more regions of the following: unstable, mask_region, power_line_default. Finally, if any Sip_defects are found inside the window, then the window is forwarded into destination defects and the function that is attached to the window at its new destination is nop (nothing to do). If no defects are found, then the window is not forwarded at all.

Figure 1. The effect of the concentration of the H_2O_2 solution on the amount of the H_2O_2 consumed in the reaction of the H_2O_2 with the H_2O_2 solution. The concentration of the H_2O_2 solution was 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 5.0, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6.0, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 7.0, 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8.0, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 9.0, 9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 9.7, 9.8, 9.9, 10.0, 10.1, 10.2, 10.3, 10.4, 10.5, 10.6, 10.7, 10.8, 10.9, 11.0, 11.1, 11.2, 11.3, 11.4, 11.5, 11.6, 11.7, 11.8, 11.9, 12.0, 12.1, 12.2, 12.3, 12.4, 12.5, 12.6, 12.7, 12.8, 12.9, 13.0, 13.1, 13.2, 13.3, 13.4, 13.5, 13.6, 13.7, 13.8, 13.9, 14.0, 14.1, 14.2, 14.3, 14.4, 14.5, 14.6, 14.7, 14.8, 14.9, 15.0, 15.1, 15.2, 15.3, 15.4, 15.5, 15.6, 15.7, 15.8, 15.9, 16.0, 16.1, 16.2, 16.3, 16.4, 16.5, 16.6, 16.7, 16.8, 16.9, 17.0, 17.1, 17.2, 17.3, 17.4, 17.5, 17.6, 17.7, 17.8, 17.9, 18.0, 18.1, 18.2, 18.3, 18.4, 18.5, 18.6, 18.7, 18.8, 18.9, 19.0, 19.1, 19.2, 19.3, 19.4, 19.5, 19.6, 19.7, 19.8, 19.9, 20.0, 20.1, 20.2, 20.3, 20.4, 20.5, 20.6, 20.7, 20.8, 20.9, 21.0, 21.1, 21.2, 21.3, 21.4, 21.5, 21.6, 21.7, 21.8, 21.9, 22.0, 22.1, 22.2, 22.3, 22.4, 22.5, 22.6, 22.7, 22.8, 22.9, 23.0, 23.1, 23.2, 23.3, 23.4, 23.5, 23.6, 23.7, 23.8, 23.9, 24.0, 24.1, 24.2, 24.3, 24.4, 24.5, 24.6, 24.7, 24.8, 24.9, 25.0, 25.1, 25.2, 25.3, 25.4, 25.5, 25.6, 25.7, 25.8, 25.9, 26.0, 26.1, 26.2, 26.3, 26.4, 26.5, 26.6, 26.7, 26.8, 26.9, 27.0, 27.1, 27.2, 27.3, 27.4, 27.5, 27.6, 27.7, 27.8, 27.9, 28.0, 28.1, 28.2, 28.3, 28.4, 28.5, 28.6, 28.7, 28.8, 28.9, 29.0, 29.1, 29.2, 29.3, 29.4, 29.5, 29.6, 29.7, 29.8, 29.9, 30.0, 30.1, 30.2, 30.3, 30.4, 30.5, 30.6, 30.7, 30.8, 30.9, 31.0, 31.1, 31.2, 31.3, 31.4, 31.5, 31.6, 31.7, 31.8, 31.9, 32.0, 32.1, 32.2, 32.3, 32.4, 32.5, 32.6, 32.7, 32.8, 32.9, 33.0, 33.1, 33.2, 33.3, 33.4, 33.5, 33.6, 33.7, 33.8, 33.9, 34.0, 34.1, 34.2, 34.3, 34.4, 34.5, 34.6, 34.7, 34.8, 34.9, 35.0, 35.1, 35.2, 35.3, 35.4, 35.5, 35.6, 35.7, 35.8, 35.9, 36.0, 36.1, 36.2, 36.3, 36.4, 36.5, 36.6, 36.7, 36.8, 36.9, 37.0, 37.1, 37.2, 37.3, 37.4, 37.5, 37.6, 37.7, 37.8, 37.9, 38.0, 38.1, 38.2, 38.3, 38.4, 38.5, 38.6, 38.7, 38.8, 38.9, 39.0, 39.1, 39.2, 39.3, 39.4, 39.5, 39.6, 39.7, 39.8, 39.9, 40.0, 40.1, 40.2, 40.3, 40.4, 40.5, 40.6, 40.7, 40.8, 40.9, 41.0, 41.1, 41.2, 41.3, 41.4, 41.5, 41.6, 41.7, 41.8, 41.9, 42.0, 42.1, 42.2, 42.3, 42.4, 42.5, 42.6, 42.7, 42.8, 42.9, 43.0, 43.1, 43.2, 43.3, 43.4, 43.5, 43.6, 43.7, 43.8, 43.9, 44.0, 44.1, 44.2, 44.3, 44.4, 44.5, 44.6, 44.7, 44.8, 44.9, 45.0, 45.1, 45.2, 45.3, 45.4, 45.5, 45.6, 45.7, 45.8, 45.9, 46.0, 46.1, 46.2, 46.3, 46.4, 46.5, 46.6, 46.7, 46.8, 46.9, 47.0, 47.1, 47.2, 47.3, 47.4, 47.5, 47.6, 47.7, 47.8, 47.9, 48.0, 48.1, 48.2, 48.3, 48.4, 48.5, 48.6, 48.7, 48.8, 48.9, 49.0, 49.1, 49.2, 49.3, 49.4, 49.5, 49.6, 49.7, 49.8, 49.9, 50.0, 50.1, 50.2, 50.3, 50.4, 50.5, 50.6, 50.7, 50.8, 50.9, 51.0, 51.1, 51.2, 51.3, 51.4, 51.5, 51.6, 51.7, 51.8, 51.9, 52.0, 52.1, 52.2, 52.3, 52.4, 52.5, 52.6, 52.7, 52.8, 52.9, 53.0, 53.1, 53.2, 53.3, 53.4, 53.5, 53.6, 53.7, 53.8, 53.9, 54.0, 54.1, 54.2, 54.3, 54.4, 54.5, 54.6, 54.7, 54.8, 54.9, 55.0, 55.1, 55.2, 55.3, 55.4, 55.5, 55.6, 55.7, 55.8, 55.9, 56.0, 56.1, 56.2, 56.3, 56.4, 56.5, 56.6, 56.7, 56.8, 56.9, 57.0, 57.1, 57.2, 57.3, 57.4, 57.5, 57.6, 57.7, 57.8, 57.9, 58.0, 58.1, 58.2, 58.3, 58.4, 58.5, 58.6, 58.7, 58.8, 58.9, 59.0, 59.1, 59.2, 59.3, 59.4, 59.5, 59.6, 59.7, 59.8, 59.9, 60.0, 60.1, 60.2, 60.3, 60.4, 60.5, 60.6, 60.7, 60.8, 60.9, 61.0, 61.1, 61.2, 61.3, 61.4, 61.5, 61.6, 61.7, 61.8, 61.9, 62.0, 62.1, 62.2, 62.3, 62.4, 62.5, 62.6, 62.7, 62.8, 62.9, 63.0, 63.1, 63.2, 63.3, 63.4, 63.5, 63.6, 63.7, 63.8, 63.9, 64.0, 64.1, 64.2, 64.3, 64.4, 64.5, 64.6, 64.7, 64.8, 64.9, 65.0, 65.1, 65.2, 65.3, 65.4, 65.5, 65.6, 65.7, 65.8, 65.9, 66.0, 66.1, 66.2, 66.3, 66.4, 66.5, 66.6, 66.7, 66.8, 66.9, 67.0, 67.1, 67.2, 67.3, 67.4, 67.5, 67.6, 67.7, 67.8, 67.9,

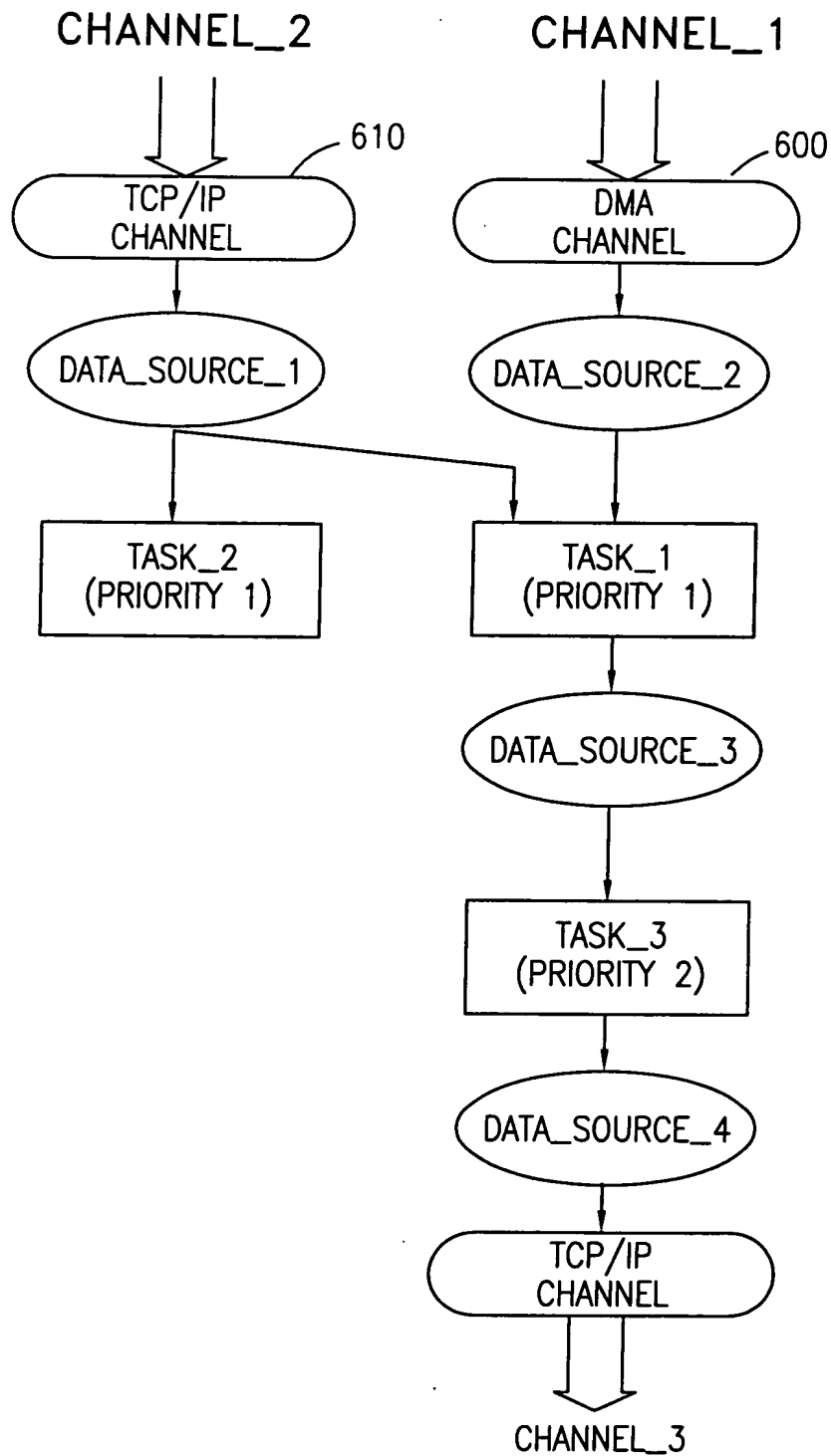
Window type	Functions attached	Remarks
reject	<p>poly compare and reject</p> <p>polylines_comp_and_reject (func_polylines_comp_aaa)</p> <p>Forward2defects (Func_forward)</p>	<p>Function polylines_comp_and reject does a CEL2VEC comparison between reference polygons and online CELs based on application target tolerances. This function has a very small limit to the number of excess CELs or missing envelopes permitted. If there is even a small change between reference target and online target overflow type defects are obtained which are interpreted by the application as an indication that this frame should be rejected.</p> <p>Finally, if any Sip_defects is found inside the window, then the window is forwarded into destination defects and the function that is attached to the window at its new destination is nop (nothing to do). If no defects are found, then the window is not forwarded at all.</p>
disqualified _lw_win	nop	Do nothing

FIG. 16

Function name	Functions executed	Remarks
balls_analyze	<p>analyze_circles (Func_circles_analyze)</p> <p>balls_compare2ref (Func_compare2ref)</p> <p>balls_display (for display and debug only. Func_display_balls_info)</p> <p>Merge_defects (Func_merge_defects)</p> <p>forward2defects (Func_forward)</p>	<p>Classify circles based on data coming from three cameras. Compare unified circles to reference. Merge all defects from cameras into main unified data.</p> <p>Finally, if any Sip defects are found inside the window, then the function is forwarded into destination defects and the function that is attached to the window as its new destination is nop (nothing to do). If no defects are found, then the window is not forwarded at all.</p>

000000" 992222999

FIG. 19



004000" 954222960

CHANNEL_3
2ND SIP PROCESS

FIG. 21A

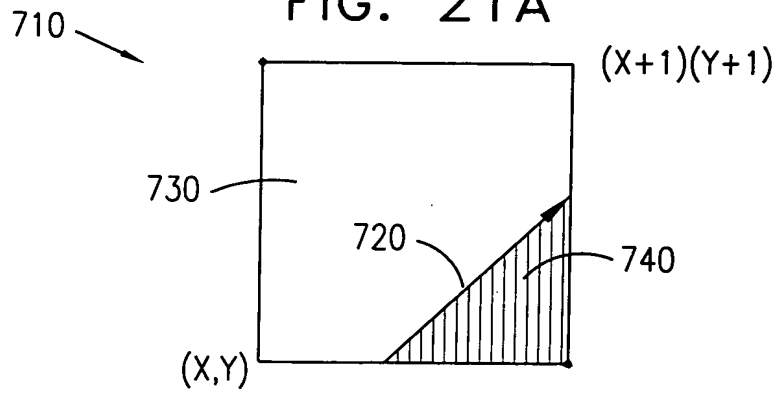


FIG. 21B

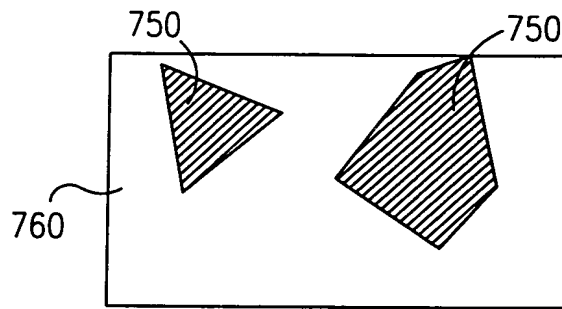


FIG. 21C

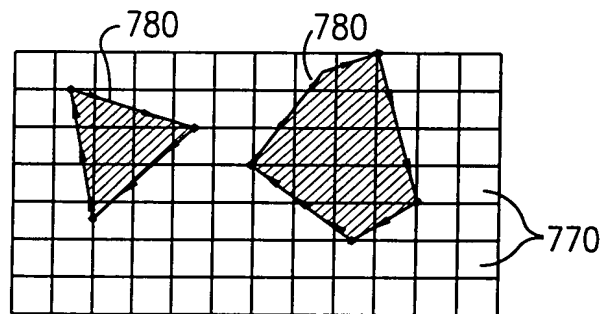


FIG. 21D

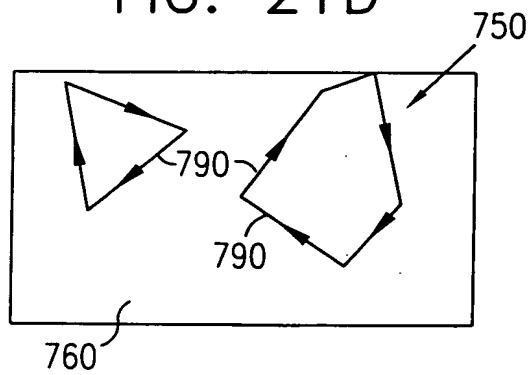


FIG. 22A

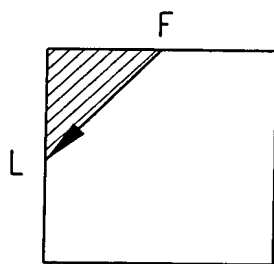


FIG. 22B

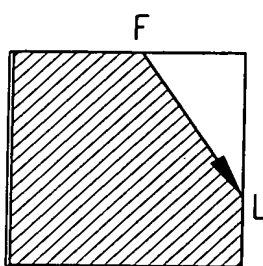


FIG. 22C

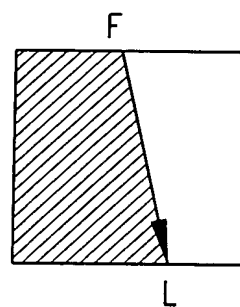


FIG. 22D

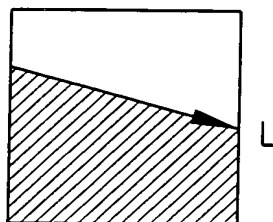


FIG. 22E

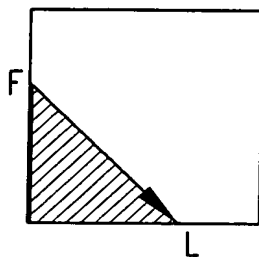


FIG. 22F

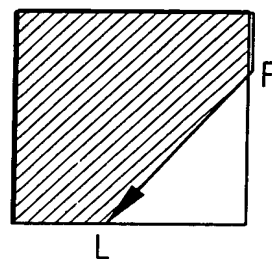


FIG. 22G

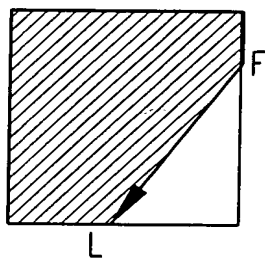
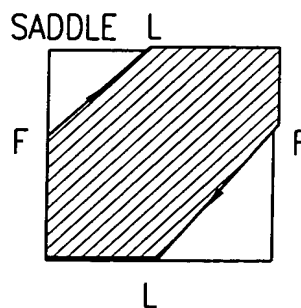


FIG. 22H



004000 93466900

004000" 95466960

FIG. 23A

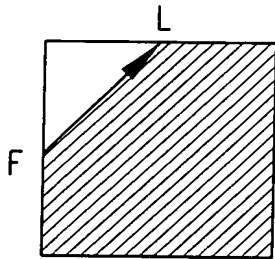


FIG. 23B

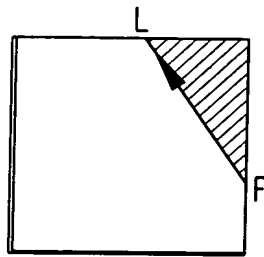


FIG. 23C

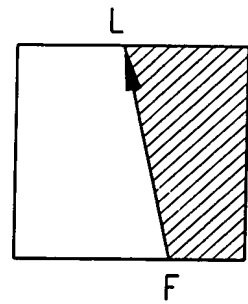


FIG. 23D

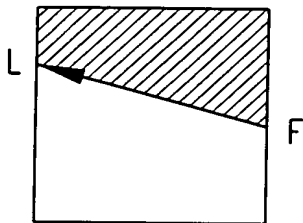


FIG. 23E

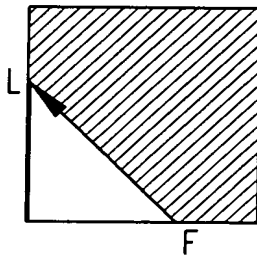


FIG. 23F

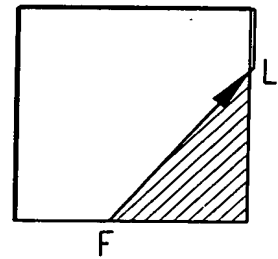


FIG. 23G

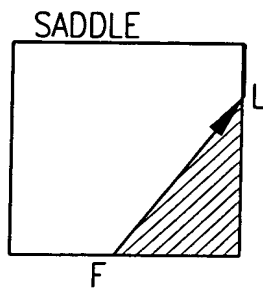


FIG. 23H

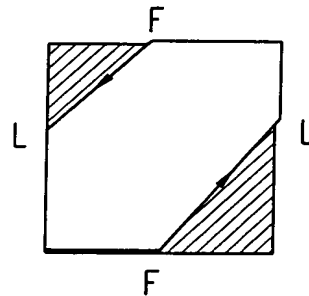


FIG. 23I

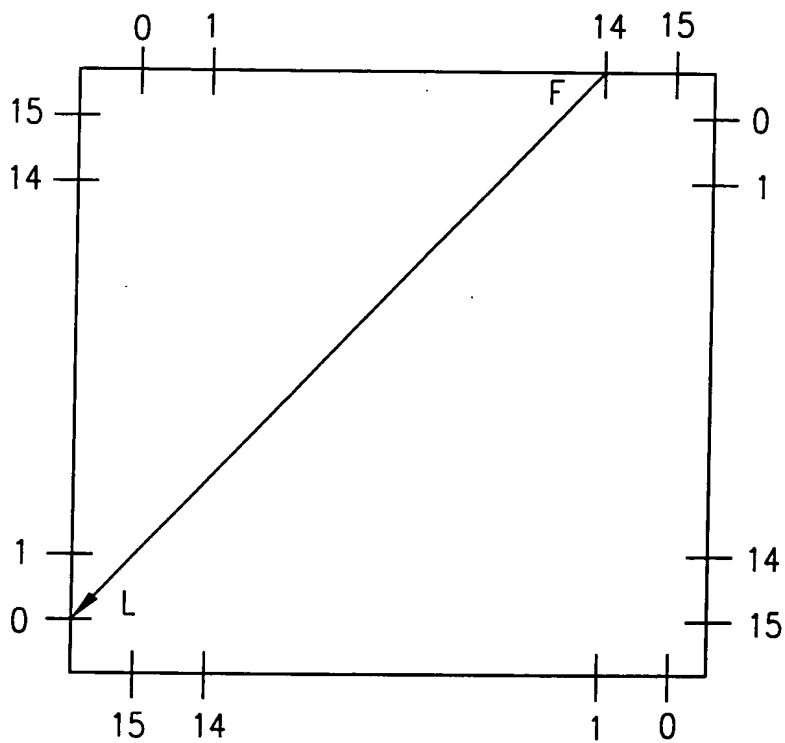
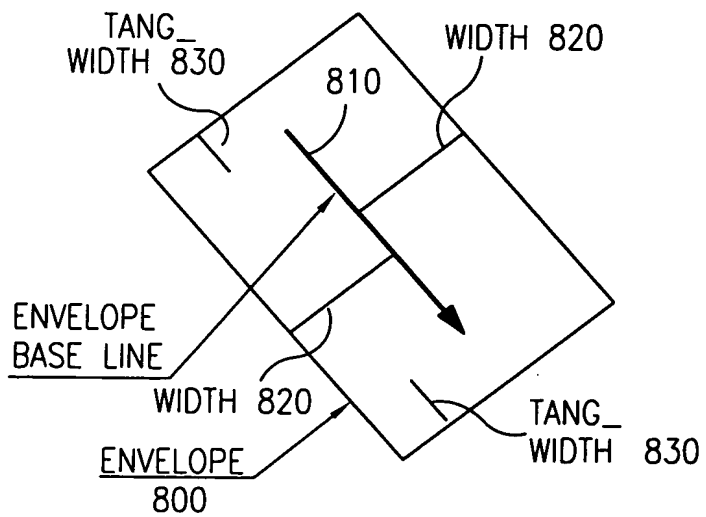


FIG. 24



004000" 95466960

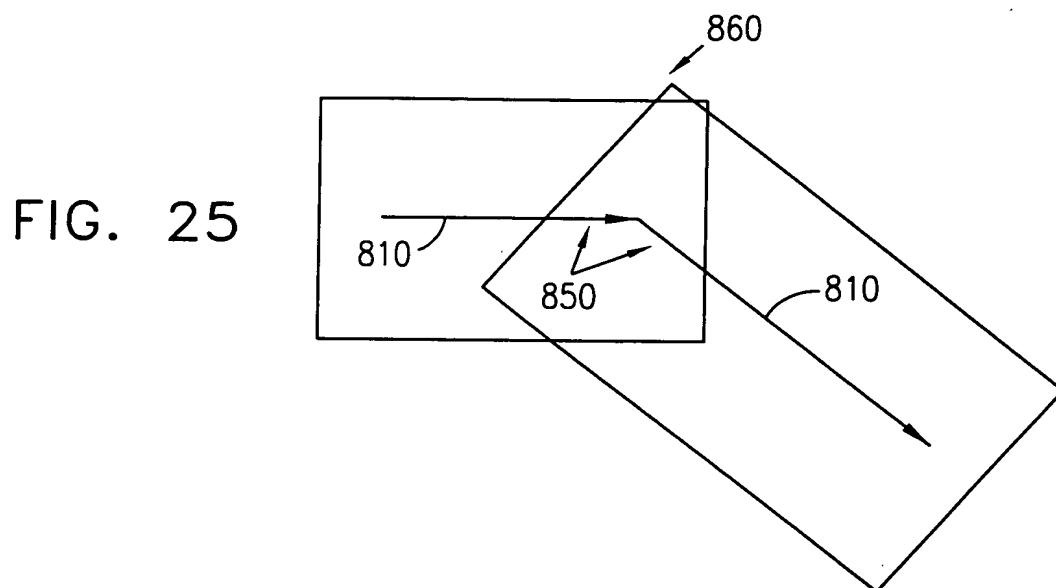


FIG. 25

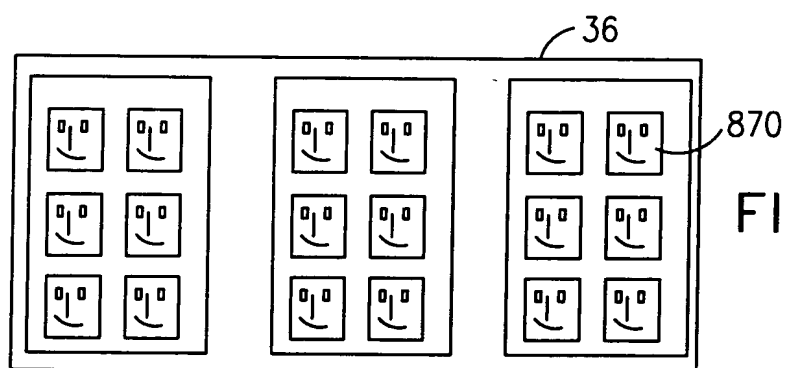


FIG. 26

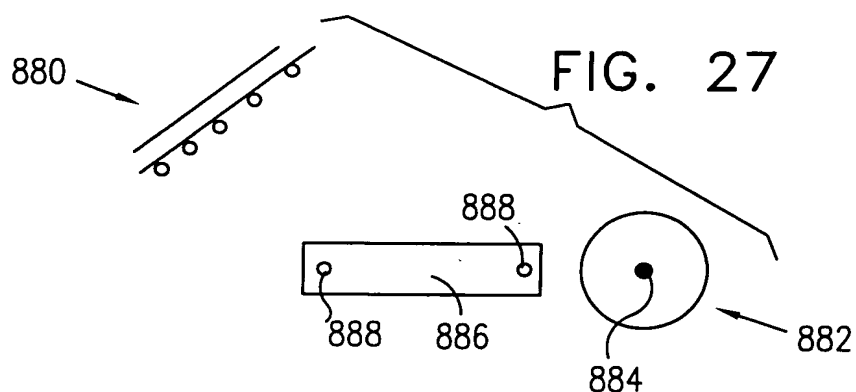


FIG. 27

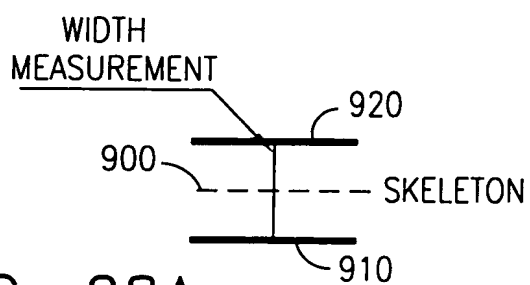


FIG. 28A

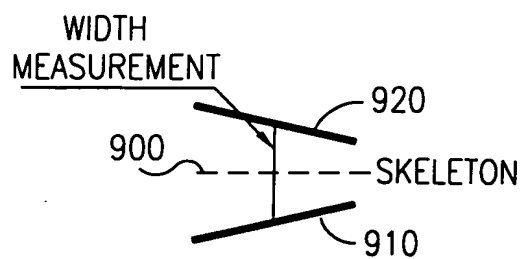
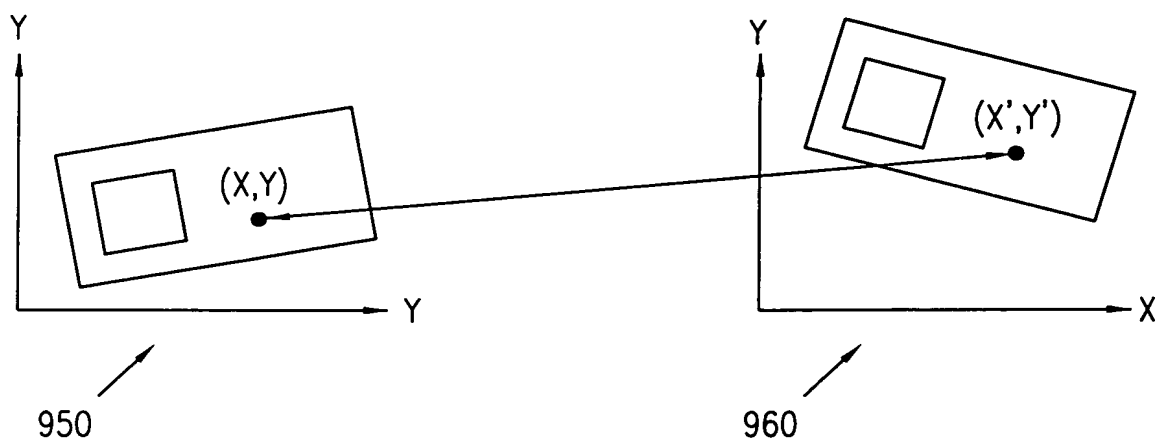
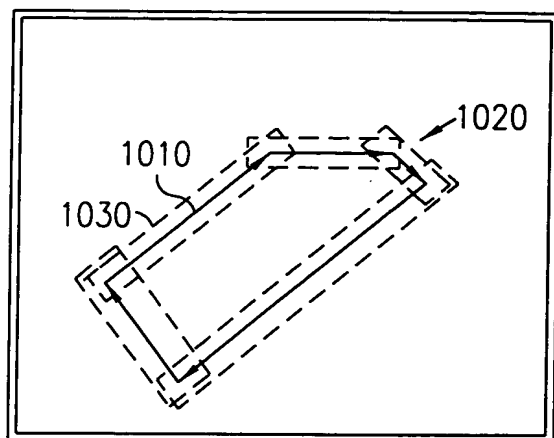


FIG. 28B

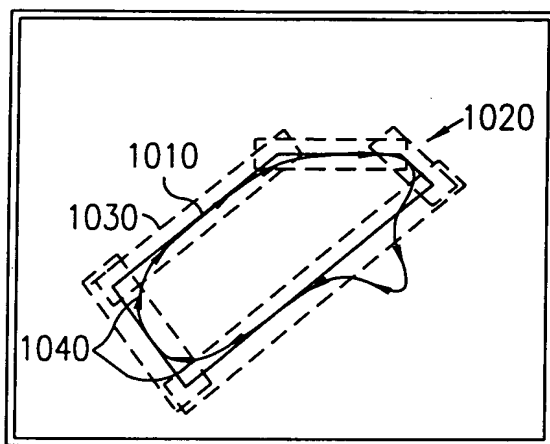
FIG. 29





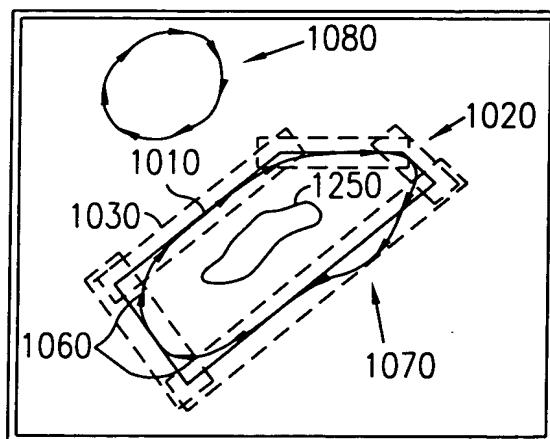
1000

FIG. 30A



1002

FIG. 30B



1004

FIG. 30C

FIG. 32A

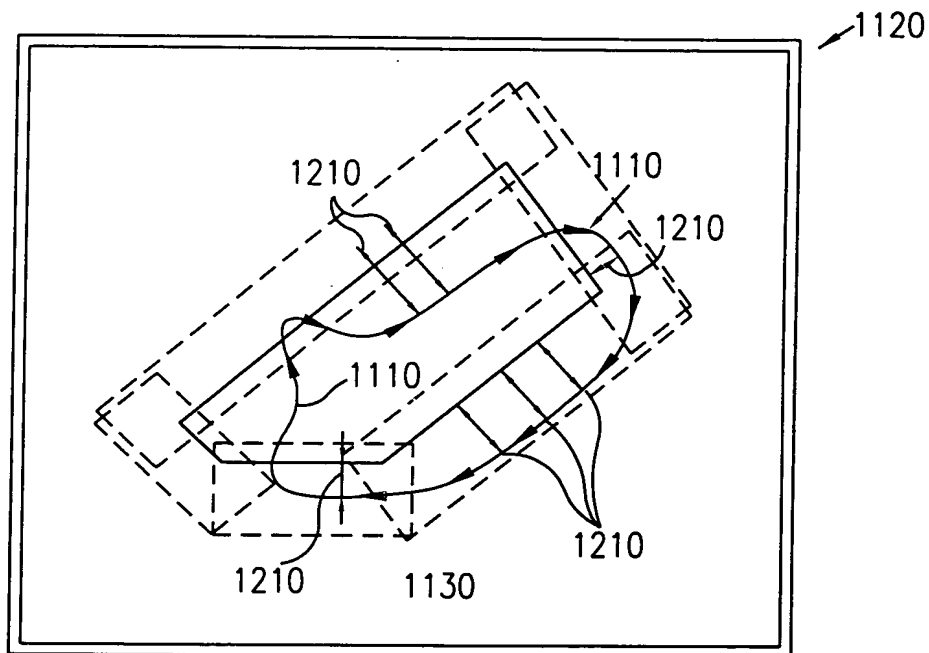


FIG. 32B

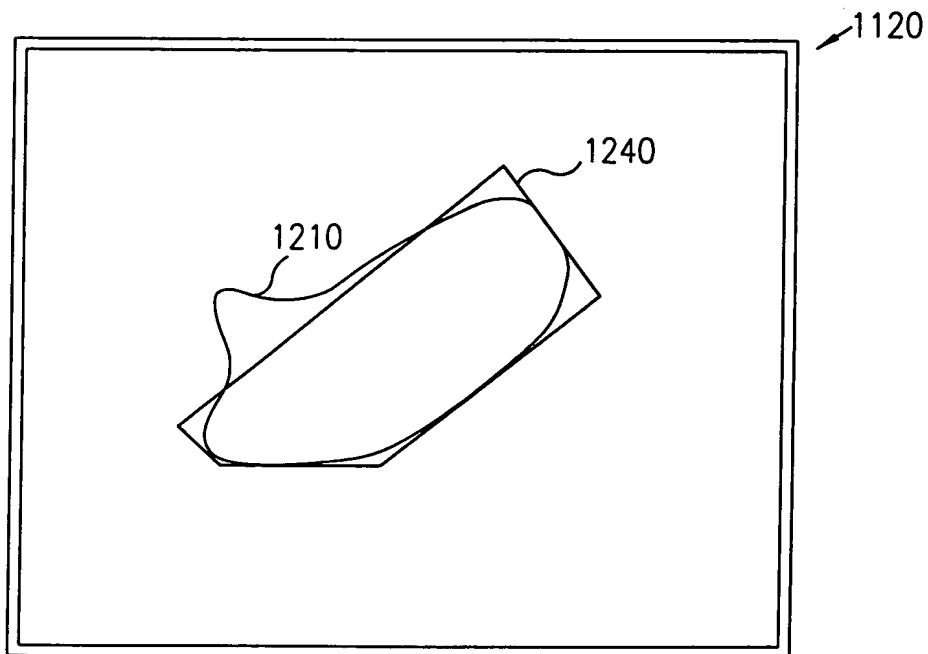
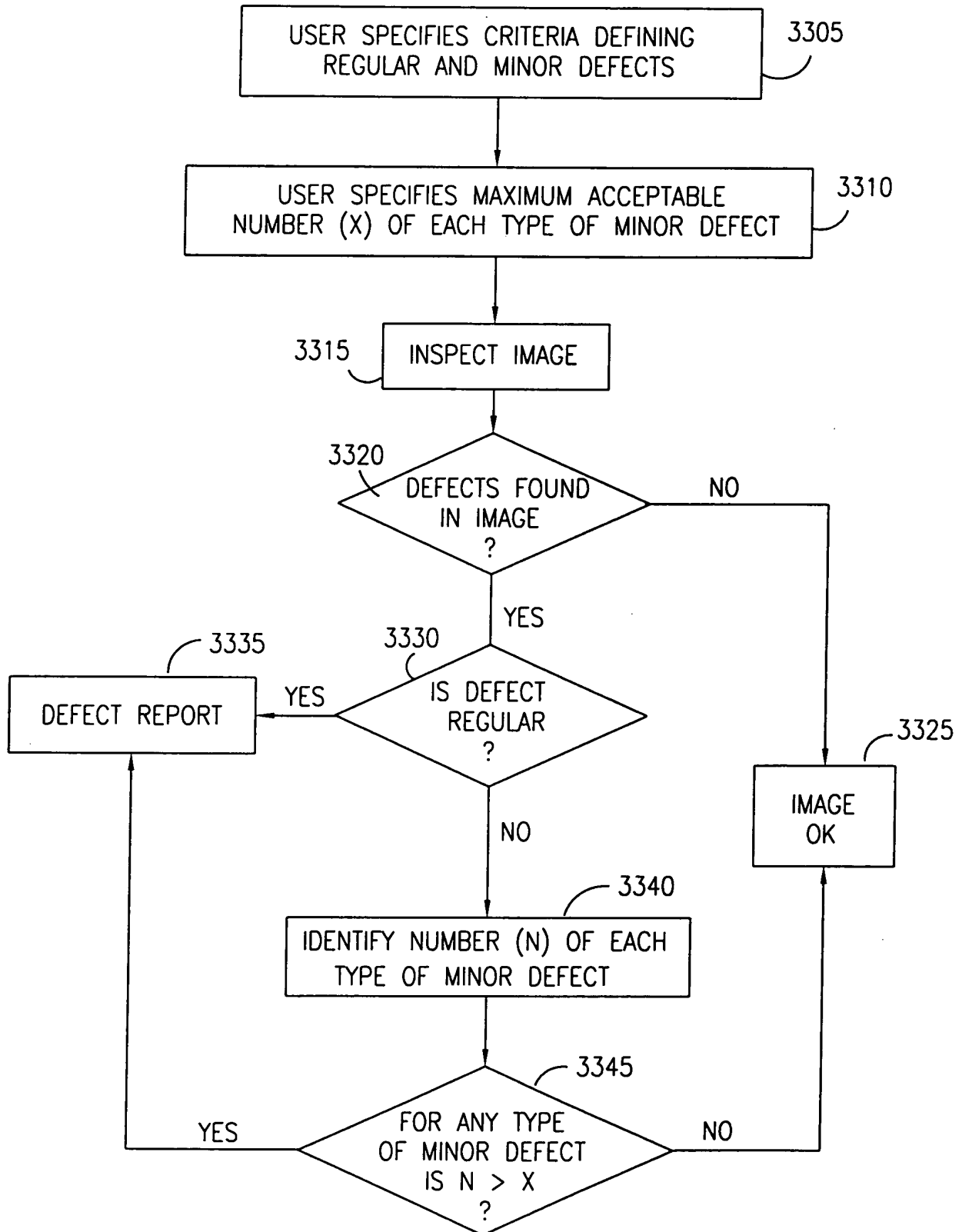


FIG. 33



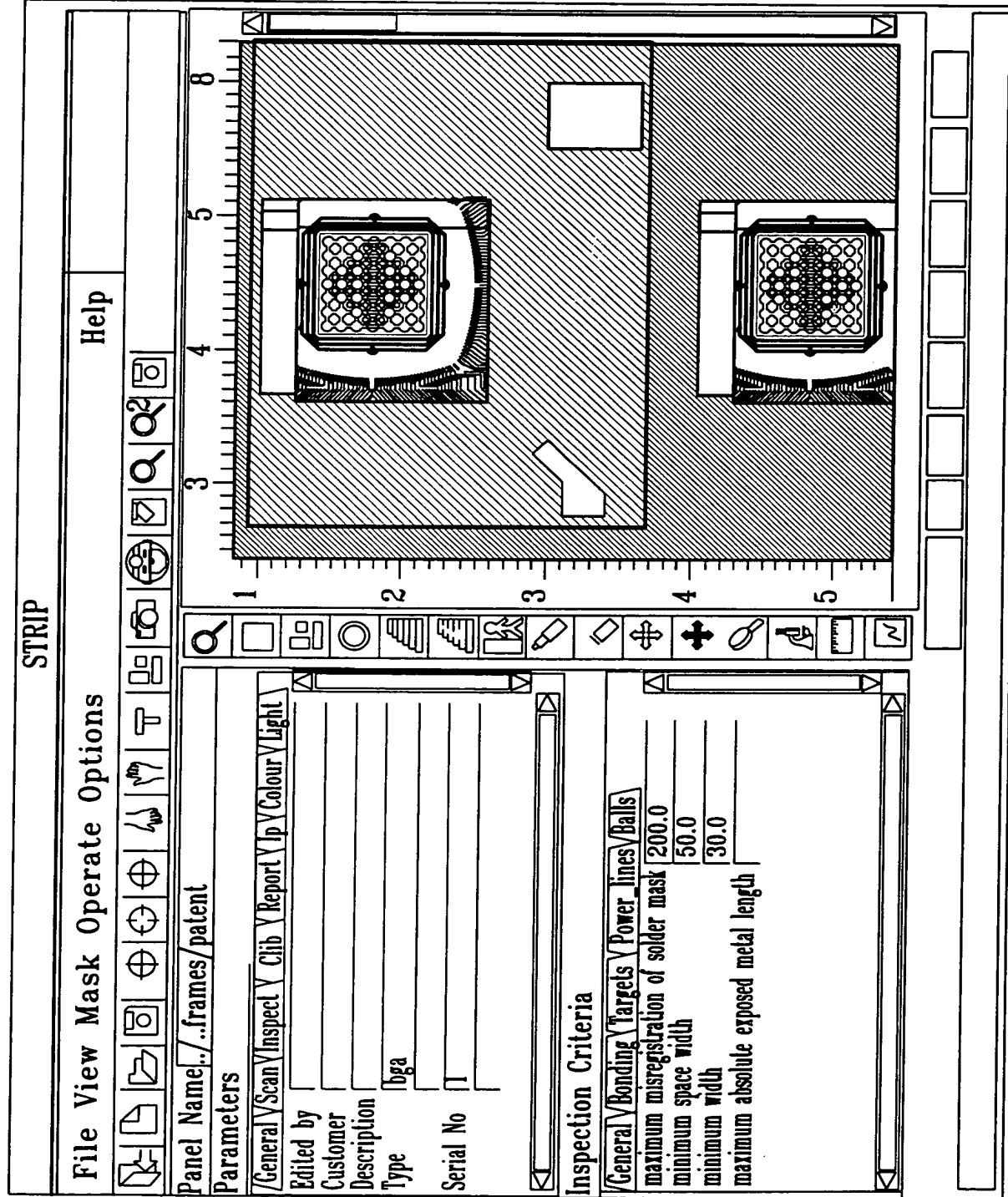


FIG. 34

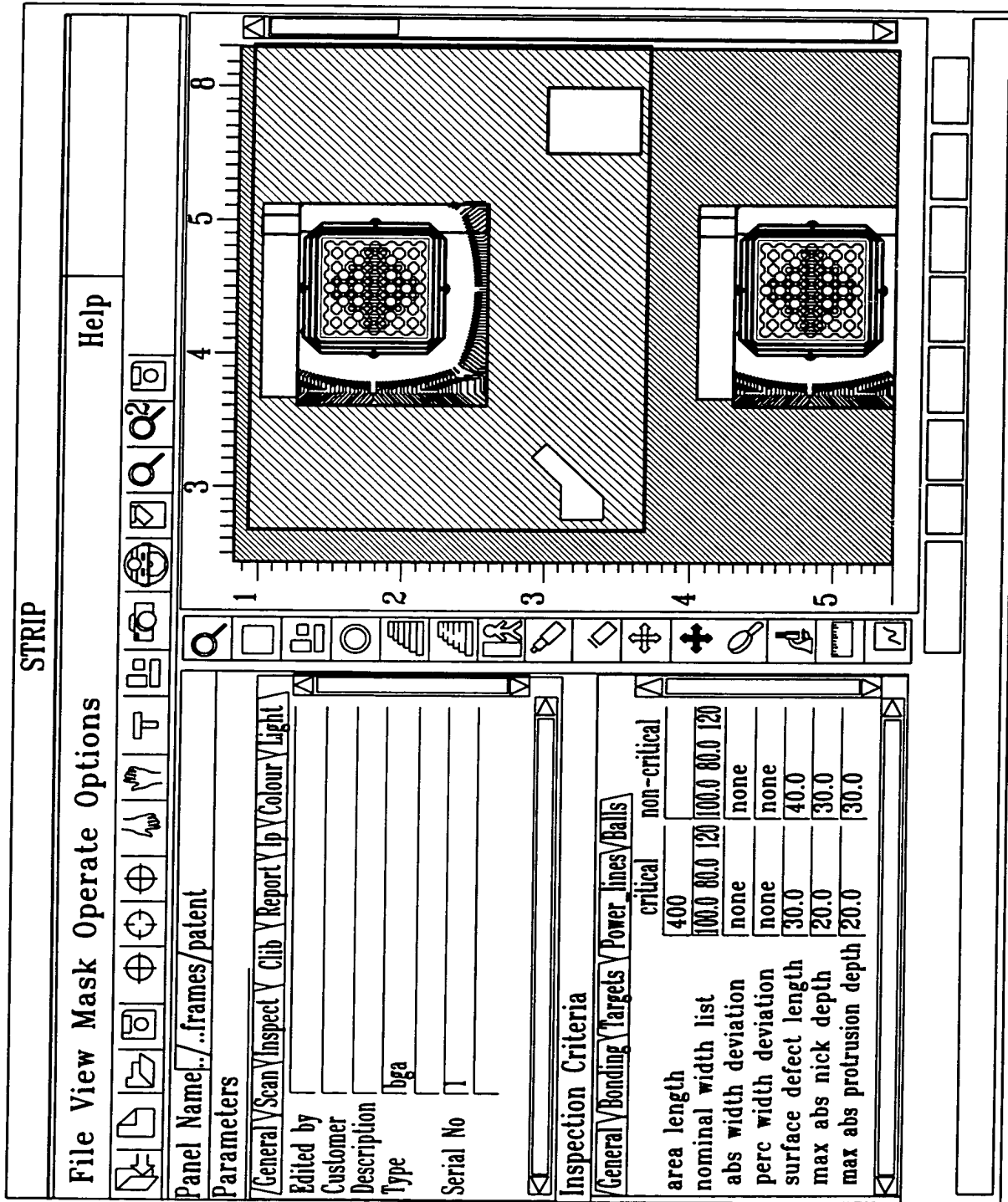


FIG. 35

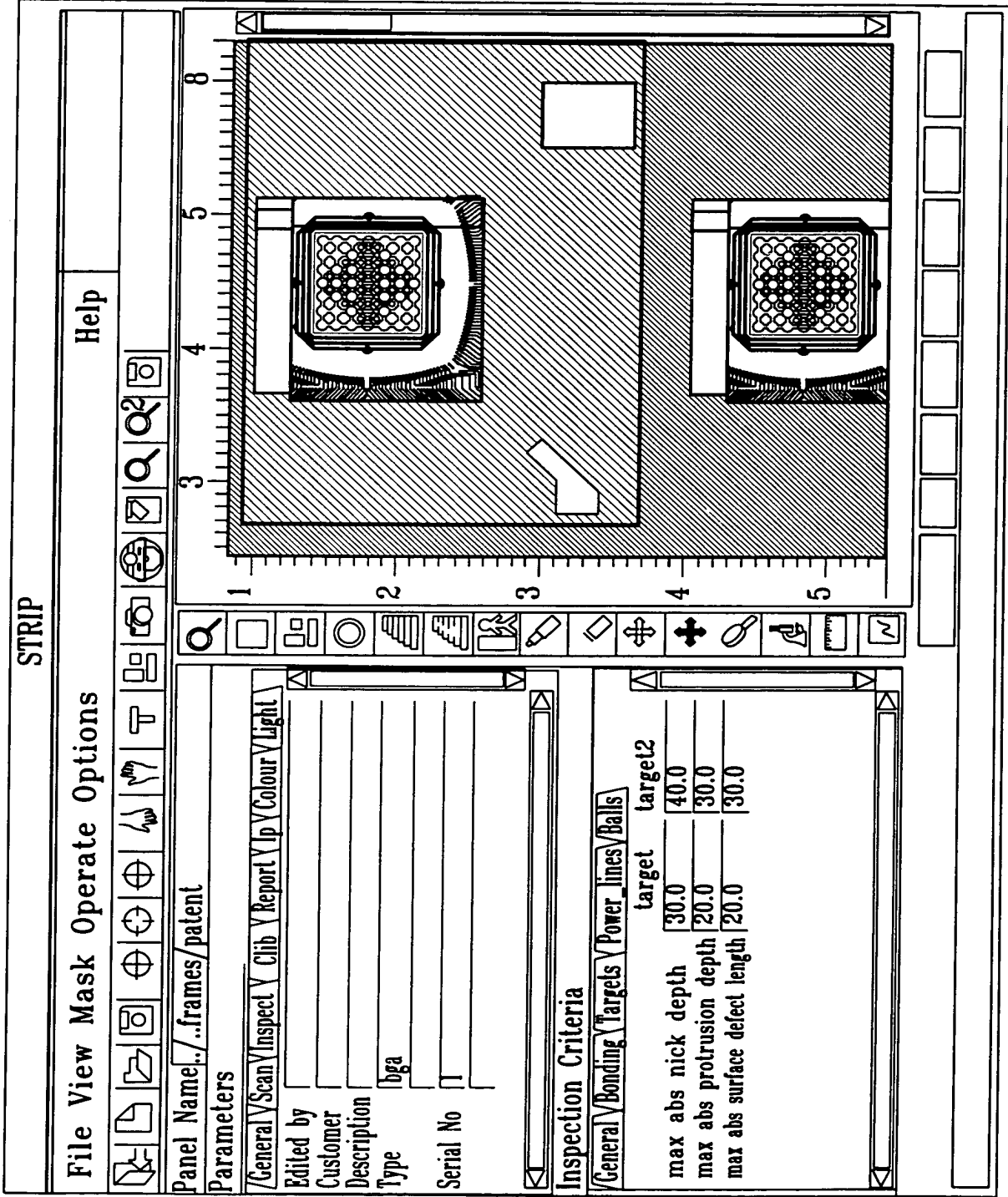


FIG. 36

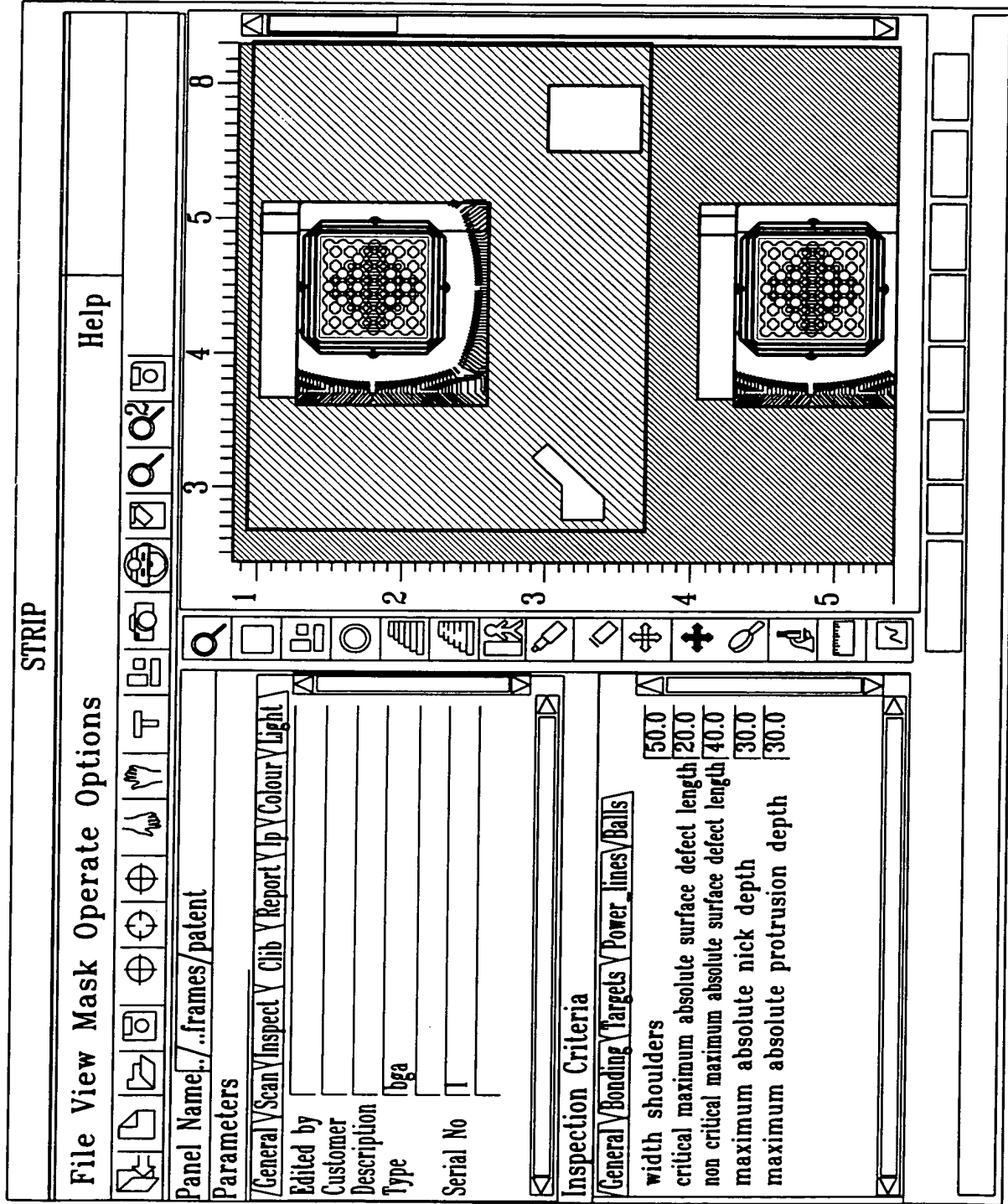


FIG. 37

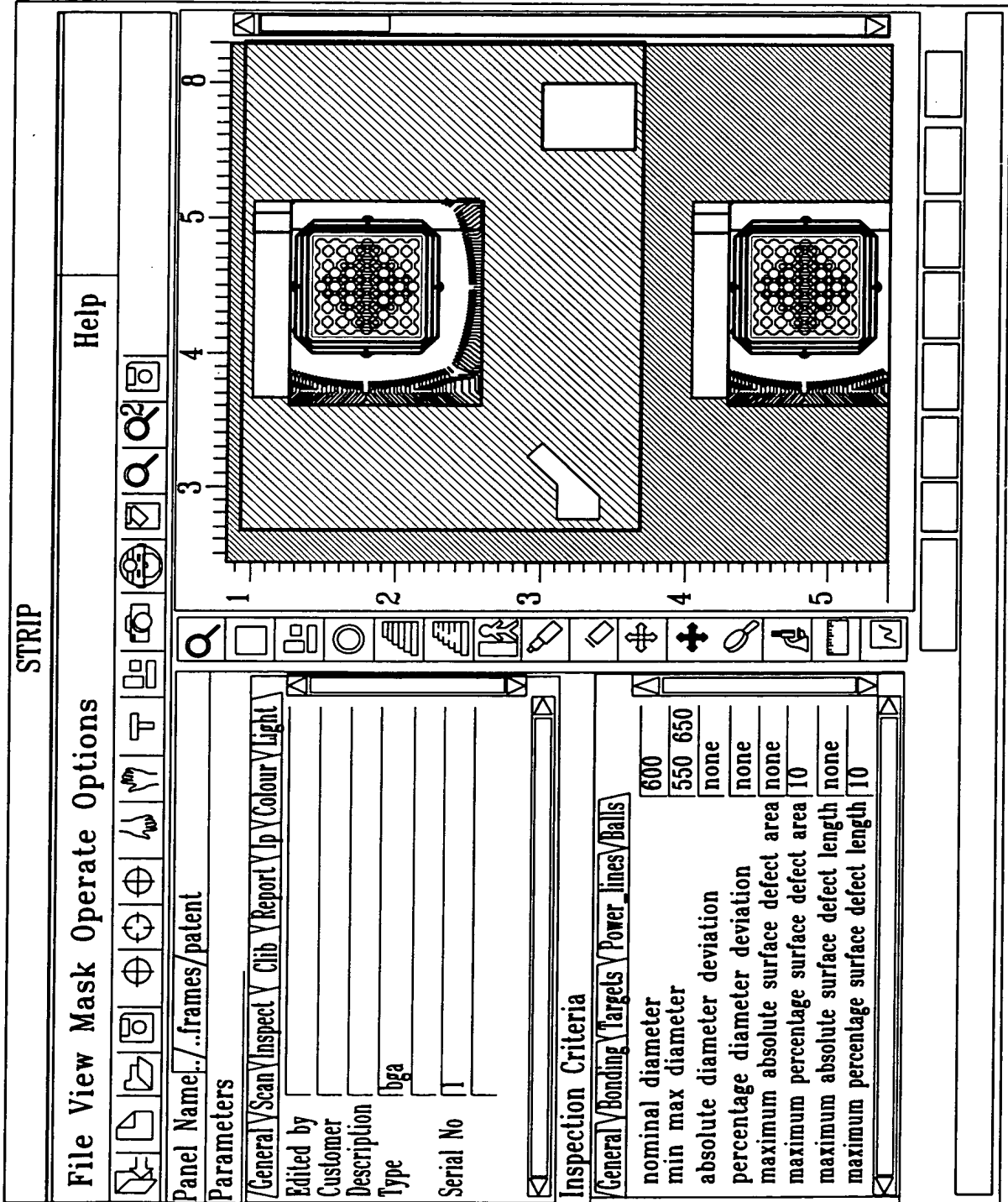


FIG. 38

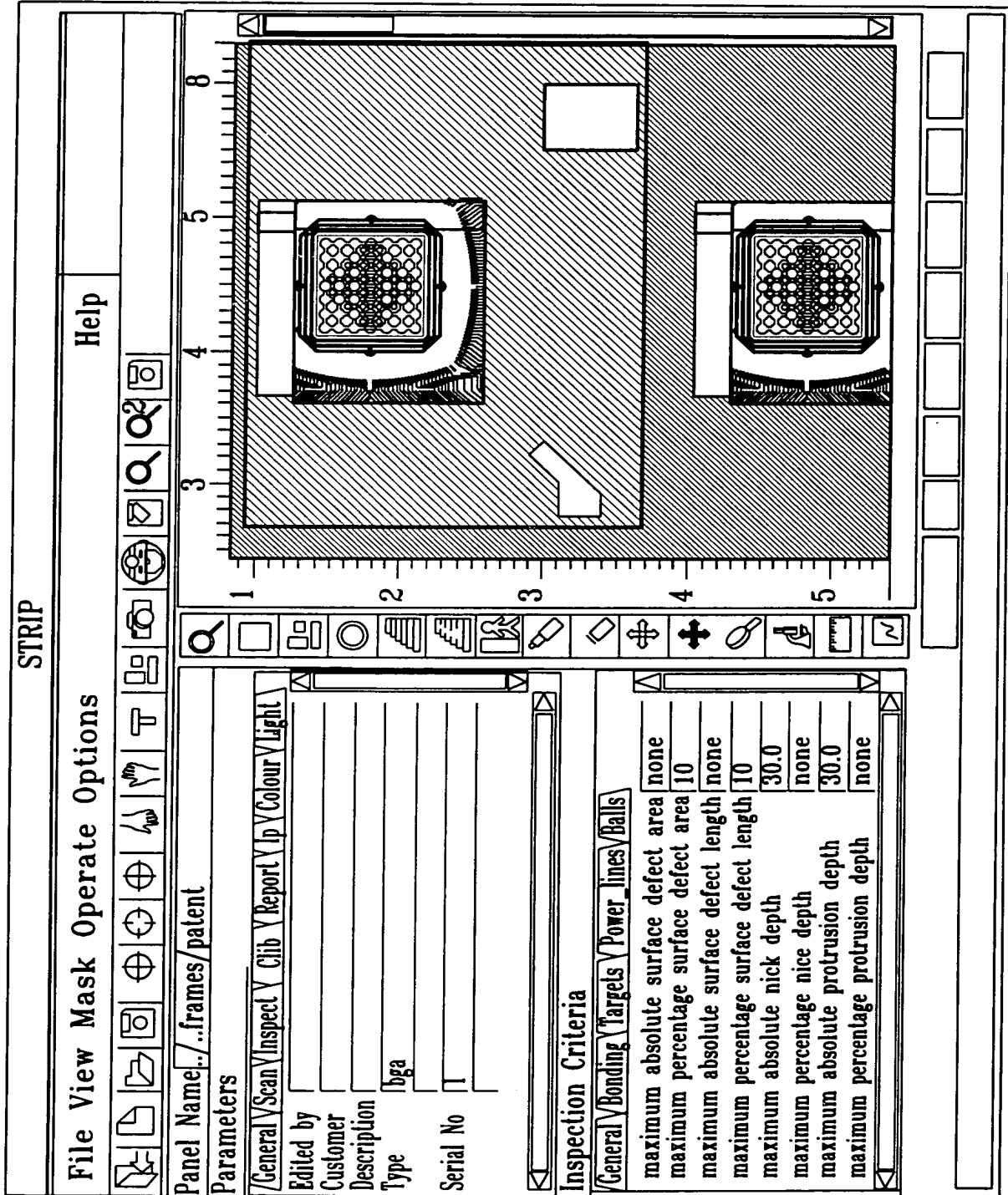


FIG. 39

FIG. 45

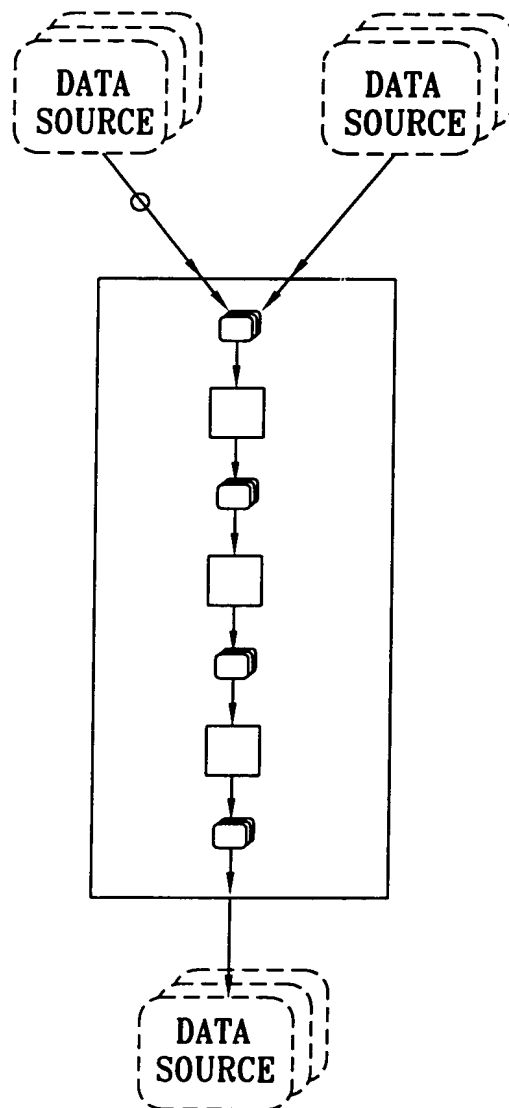
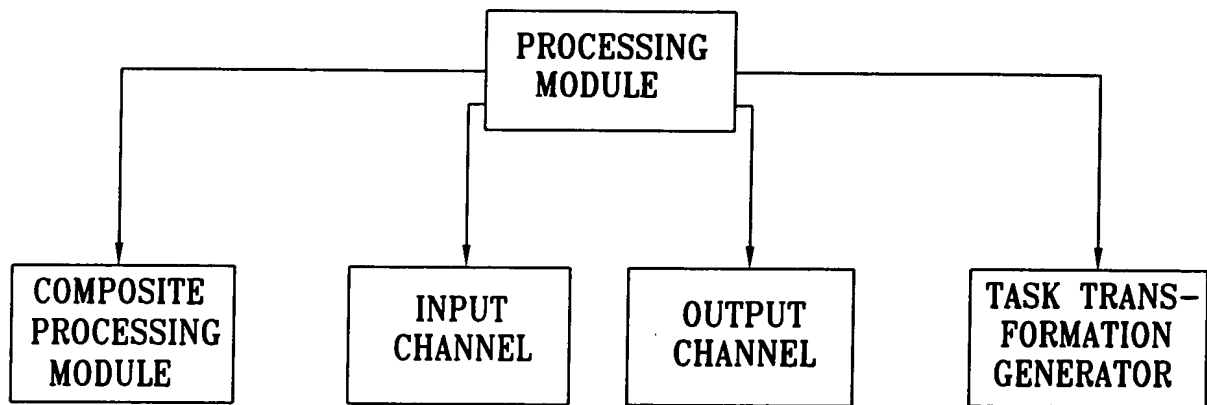
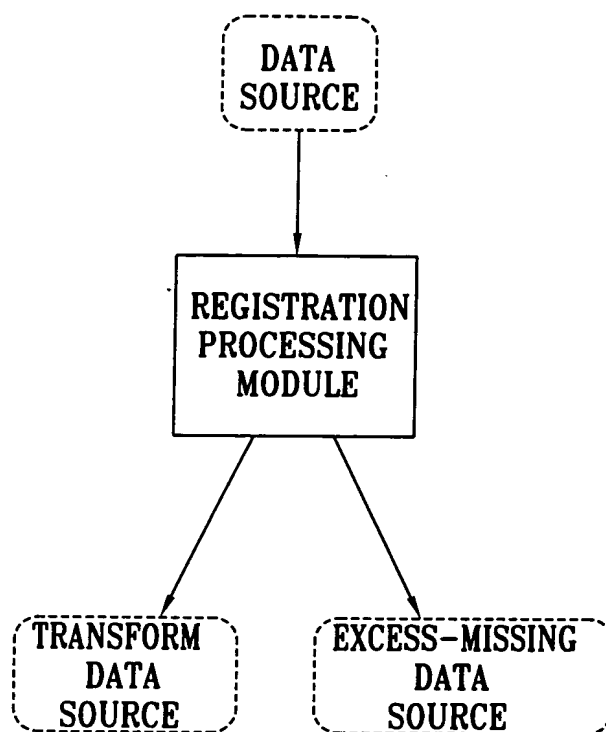


FIG. 46

FIG. 49



09637EE.080700 000000" 95422960

FIG. 50

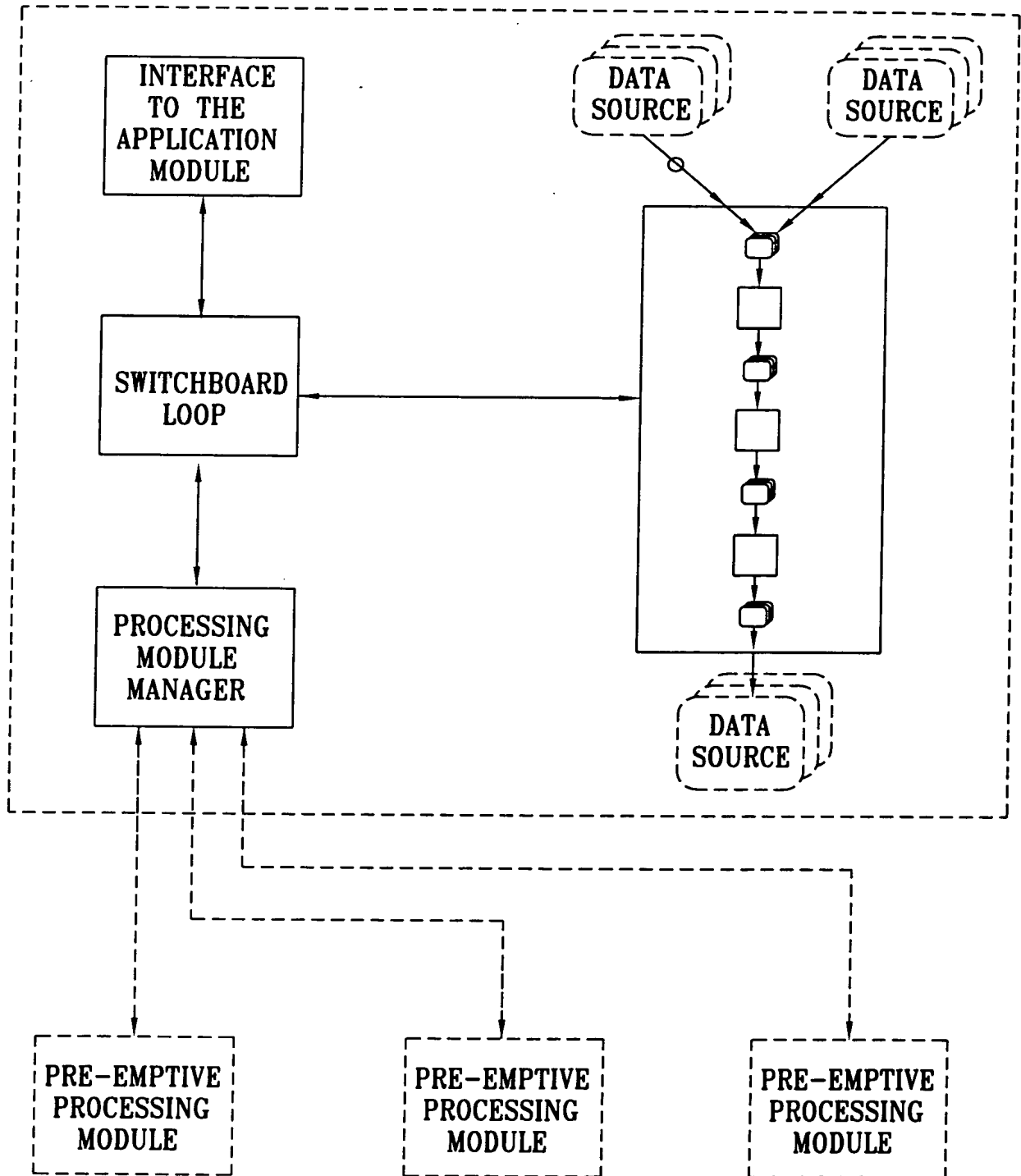
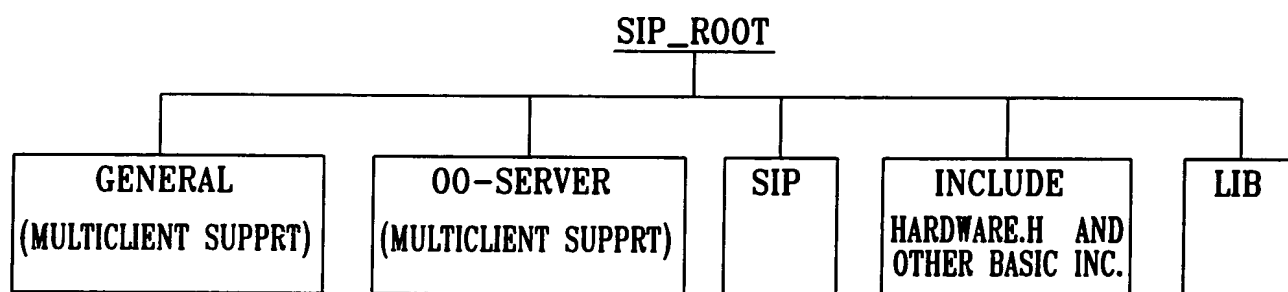


FIG. 52



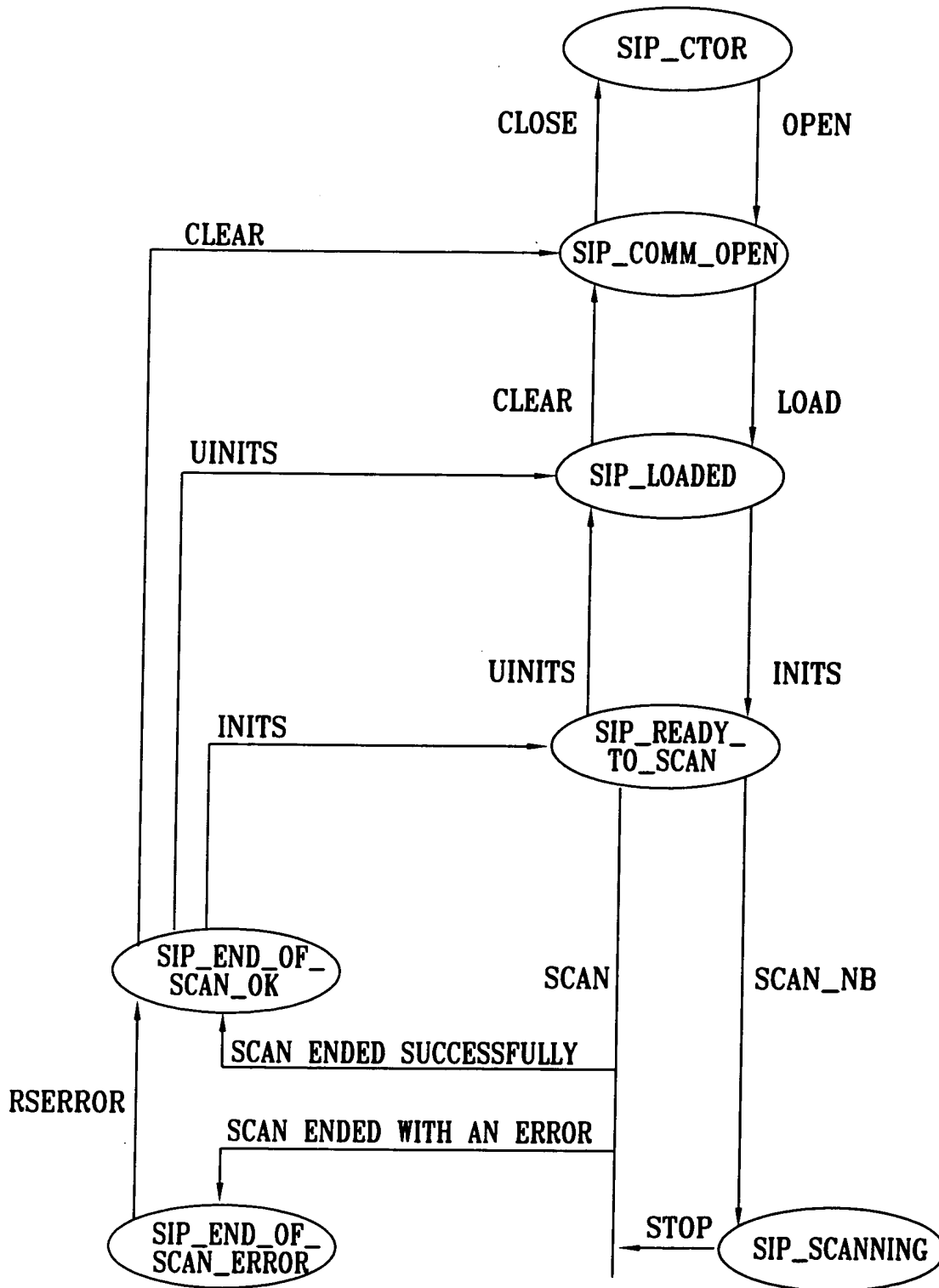
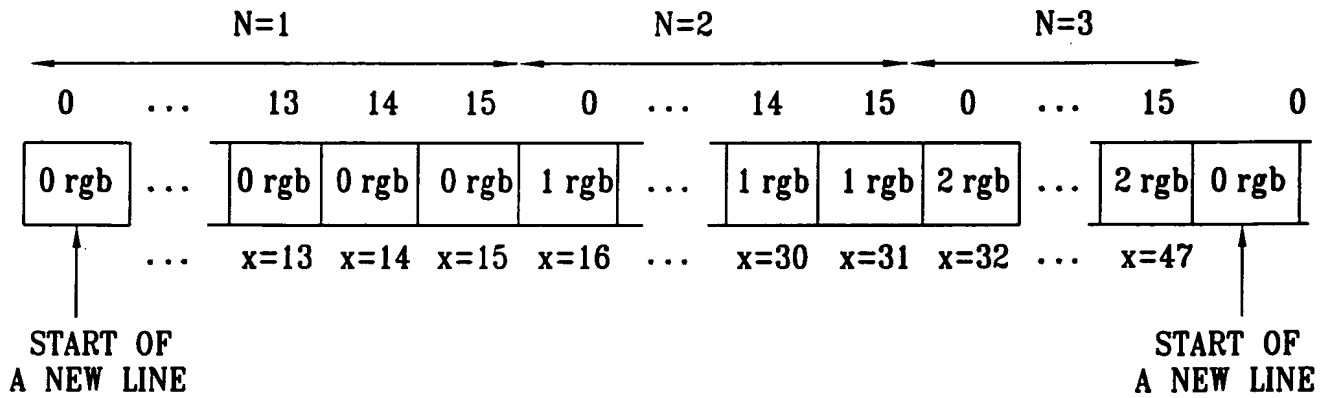
[illegible]

FIG. 56

$x_s \text{ rgb}$

REPRESENTS A SINGLE 32-BIT SNAP REPORT WORD. WHICH CONTAINS THE 8-BIT X-COORDINATE (x_s) AND THE VALUES OF THE RED (r) GREEN (g) AND BLUE (b)



METAL COVERED BY SOLDER MASK

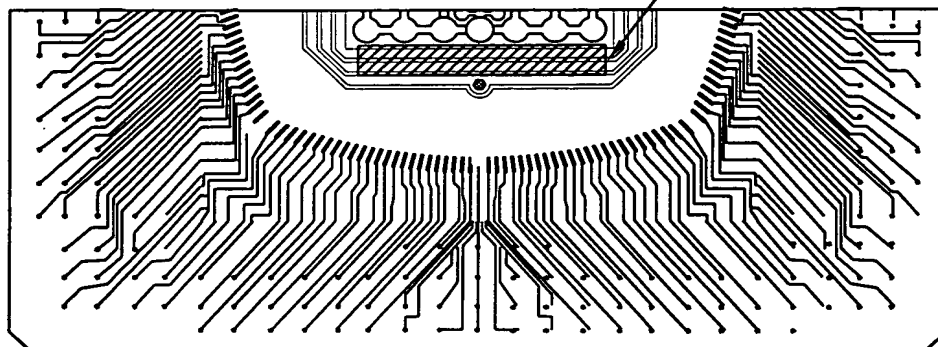
METAL REGIONS

METAL REGIONS

FIG. 57

METAL COVERED BY SOLDER MASK

FIG. 58 REGION TO BE IGNORED



004000" 954222960

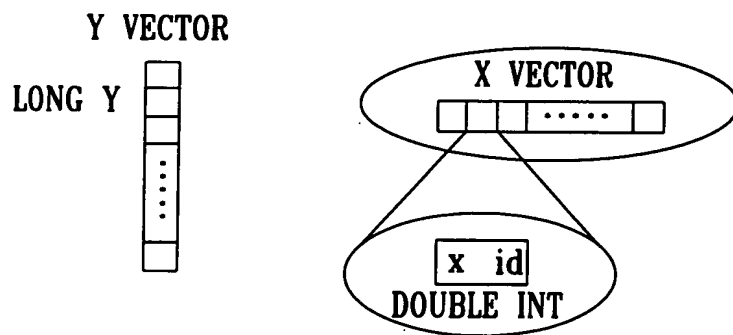


FIG. 62

WHERE THE GRAPHIC INTERPRETATION IS:

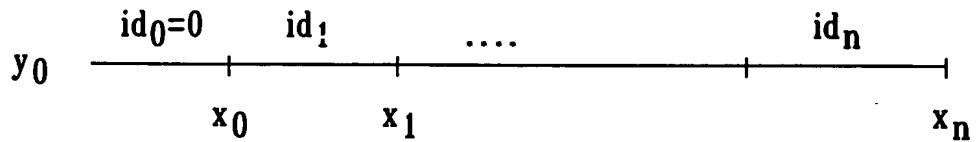


FIG. 63

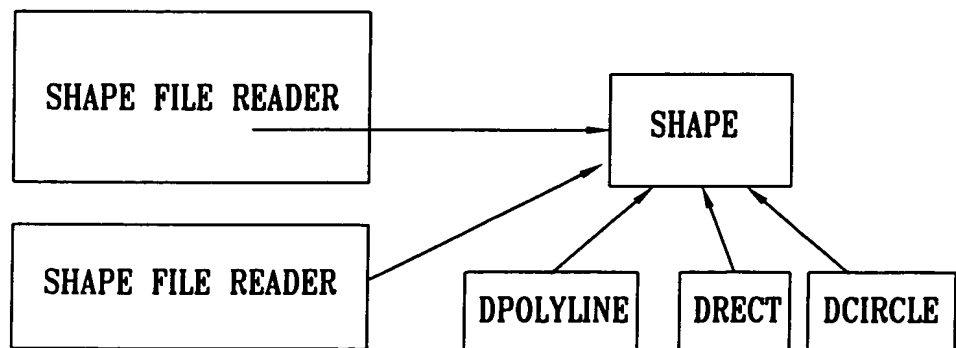
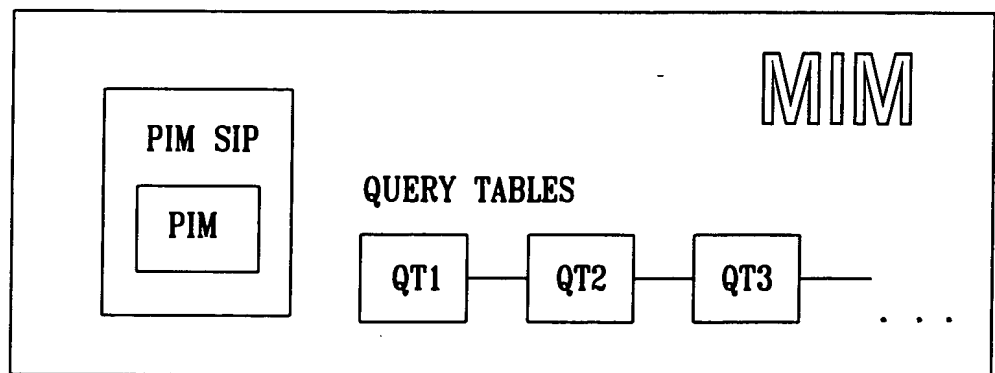


FIG. 64

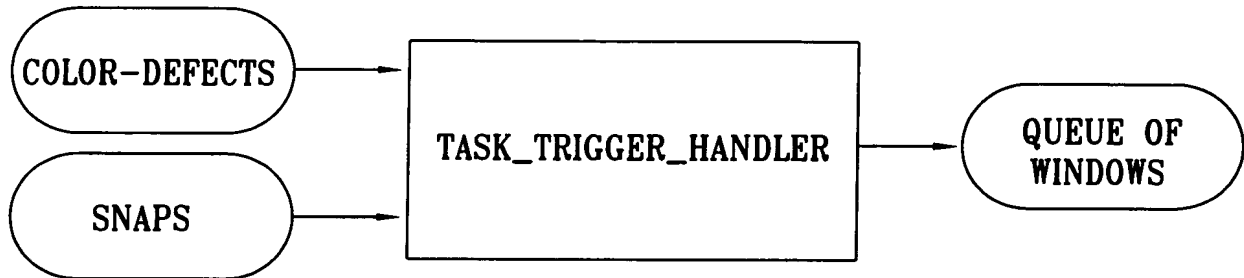
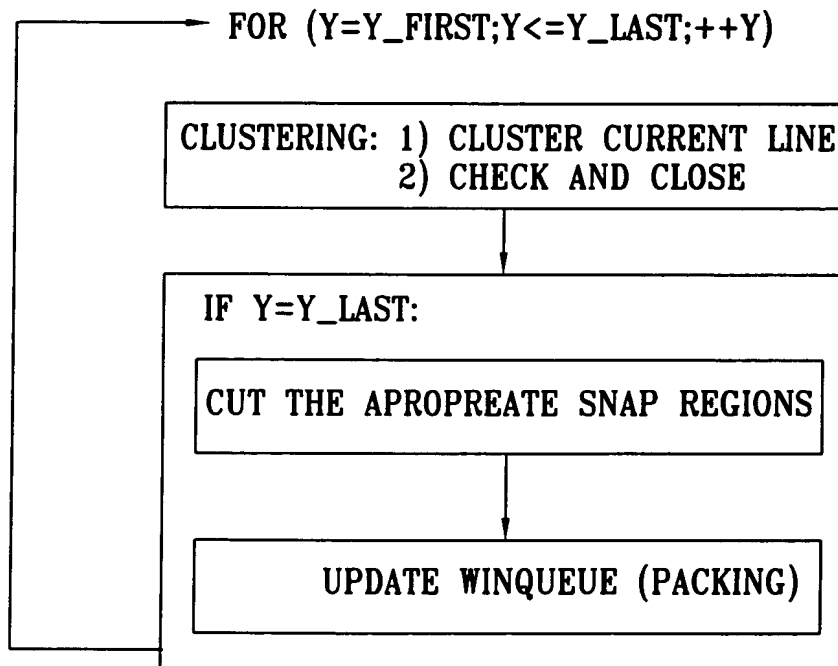


FIG. 65



[illegible]

Y=5	10'o'1	15'c'1						
Y=6	10'o'1	15'c'1	22'o'2	26'o'2				
Y=7	10'o'1	15'c'1	22'o'2	25'o'3	26'c'2	29'c'3		
Y=8	10'o'1	15'c'1	22'o'2	25'o'3	26'c'2	29'c'3		
Y=9	10'o'1	15'c'1	22'o'2	25'o'3	26'c'2	29'c'3		
Y=10	10'o'1	15'c'1	16'o'4	20'c'4	22'o'2	25'o'3	26'c'2	29'c'3
Y=11	10'o'1	15'c'1	16'o'4	20'c'4	22'o'2	25'o'3	26'c'2	29'c'3
Y=12	16'o'4	20'c'4	25'o'3	29'c'3				
Y=13	16'o'4	20'c'4						
Y=14	16'o'4	20'c'4						
...								

FIG. 67A

THE SNAP RECTANGULAR REGIONS (COLORED AREAS)
THAT WERE CUT ACCORDING TO THE CLUSTERS

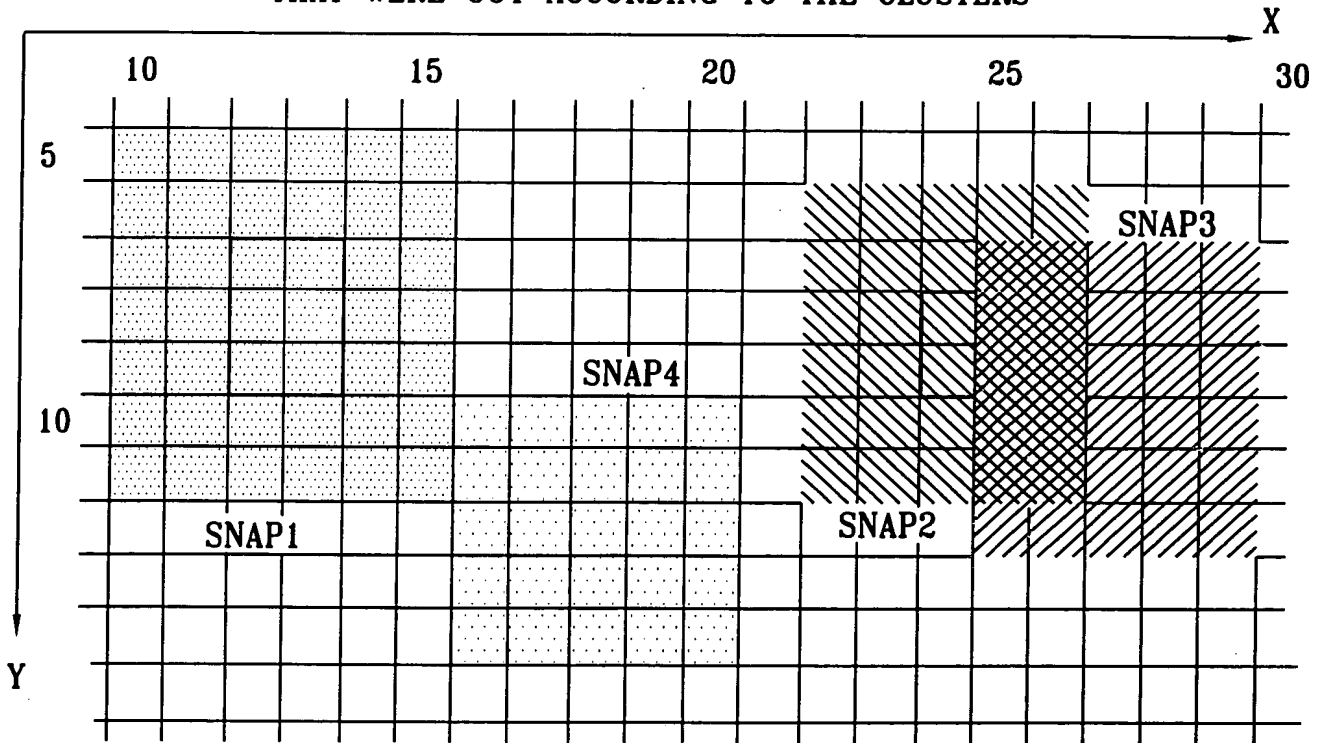
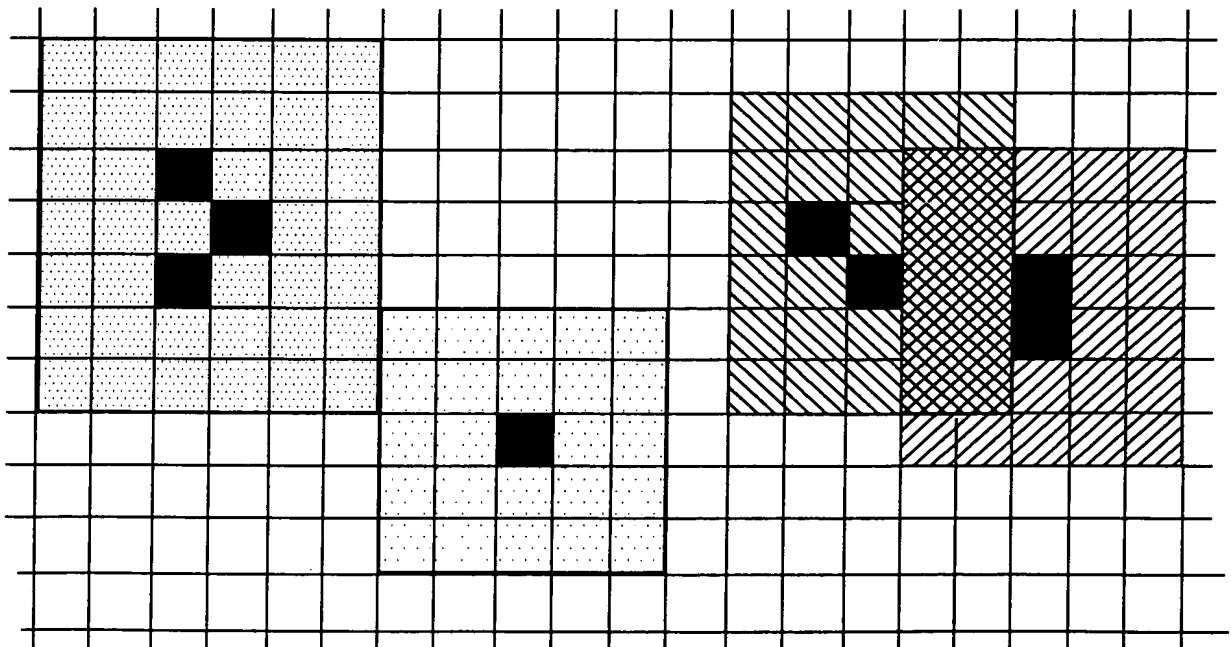


FIG. 67B

THE RESULT IS 4 WINDOWS, EACH POSSESS THE CLUSTER
AND A RECTANGLE WITH THE RED-DATA.



090726Z JUL 80

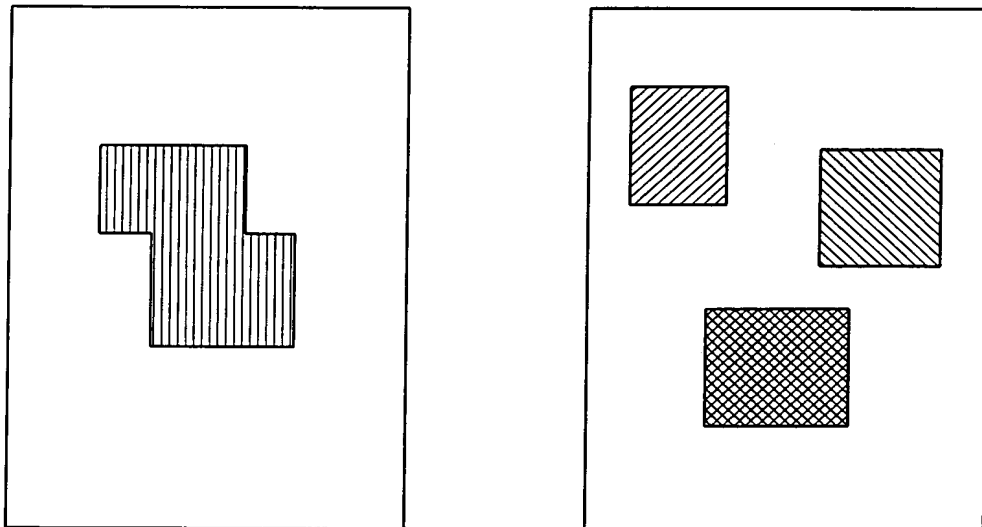


FIG. 70

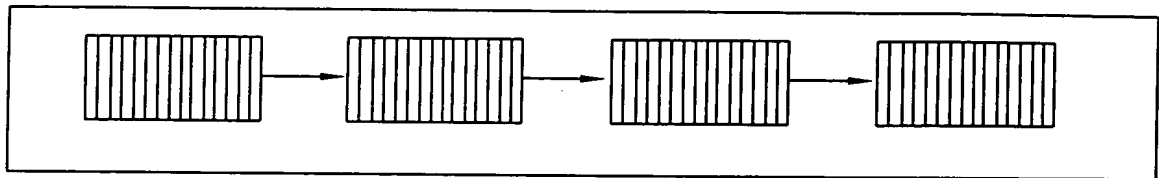


FIG. 71

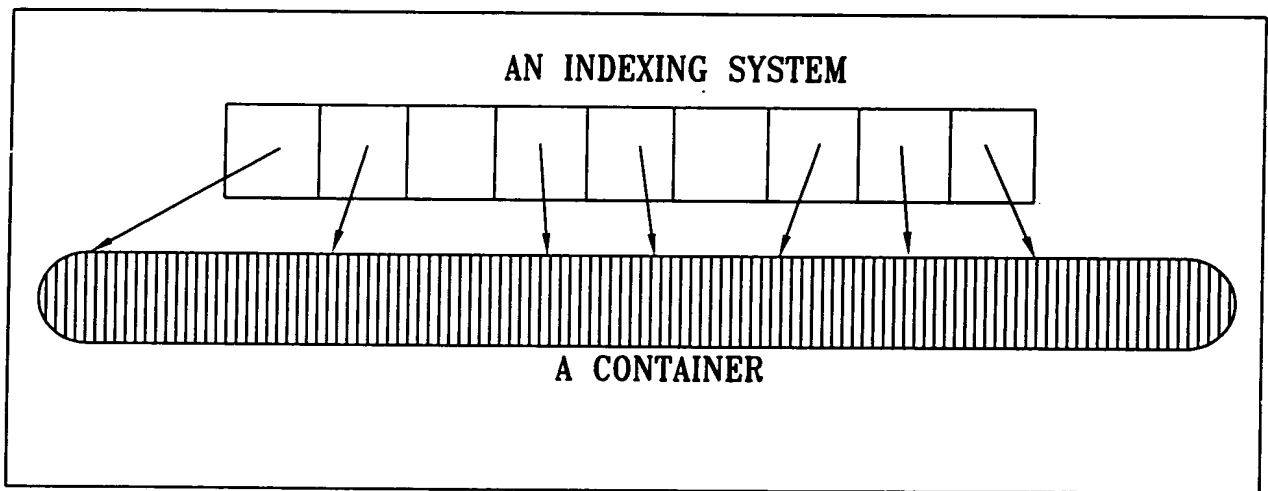
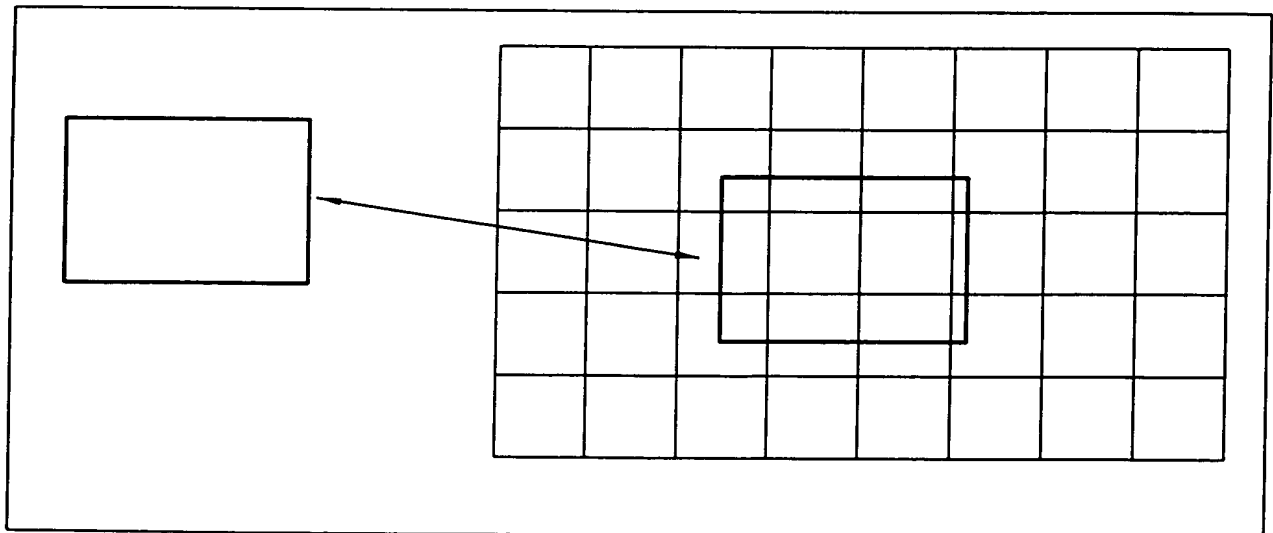


FIG. 72



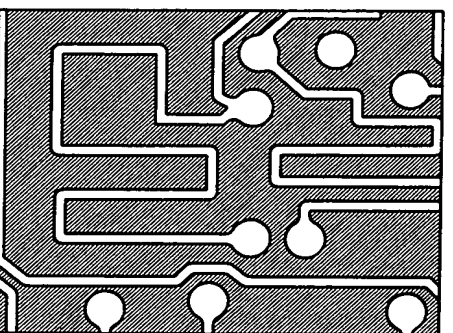
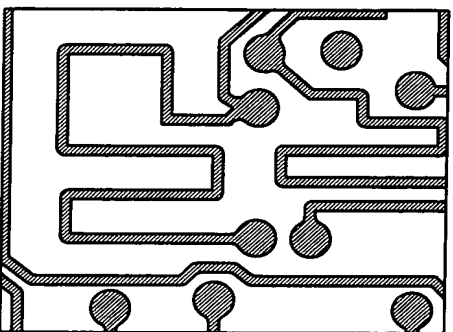
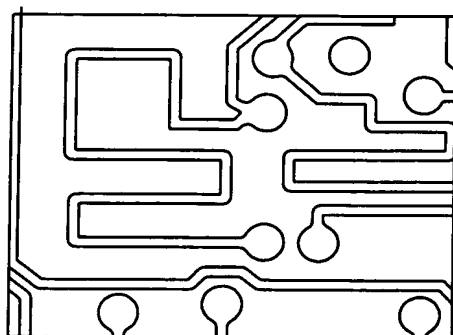
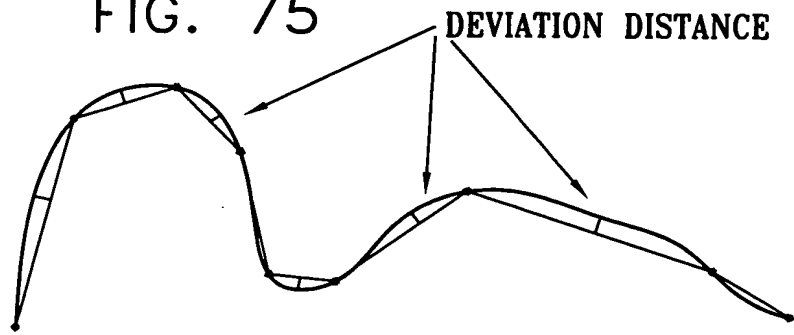
[illegible]

FIG. 74

FIG. 75



LOW FREQUENCY
DEFLECTION

FIG. 76

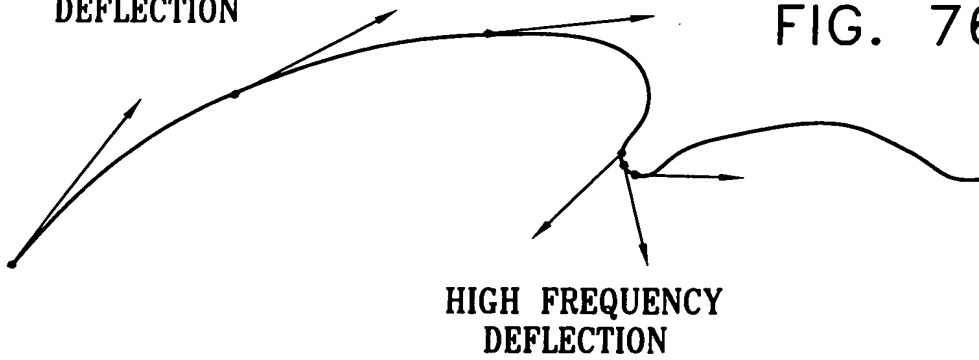
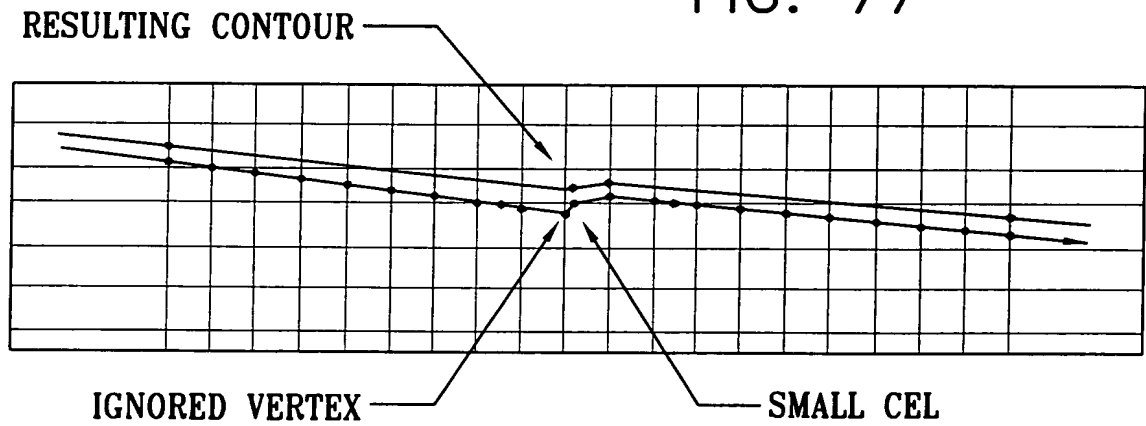


FIG. 77



MAXIMUM DISTANCE
FROM ORIGINAL CONTOUR

FIG. 78

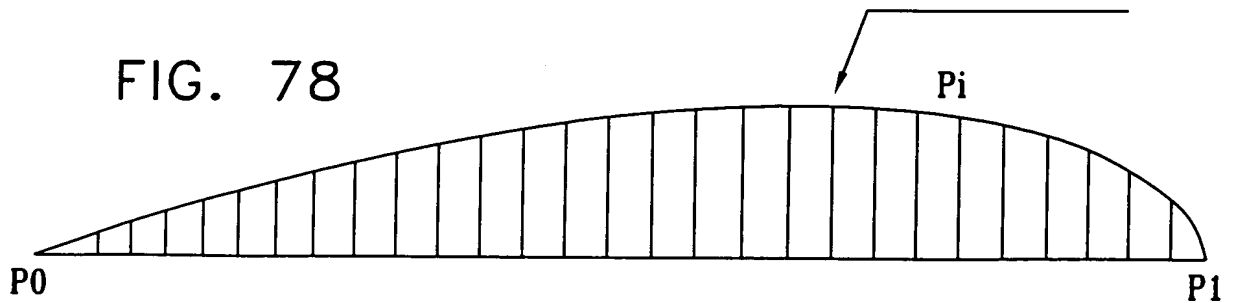


FIG. 79

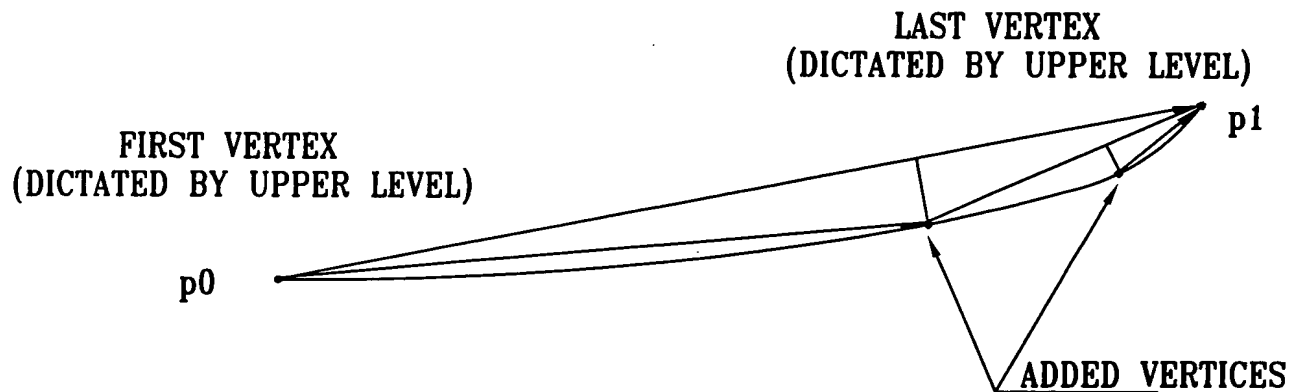


FIG. 80

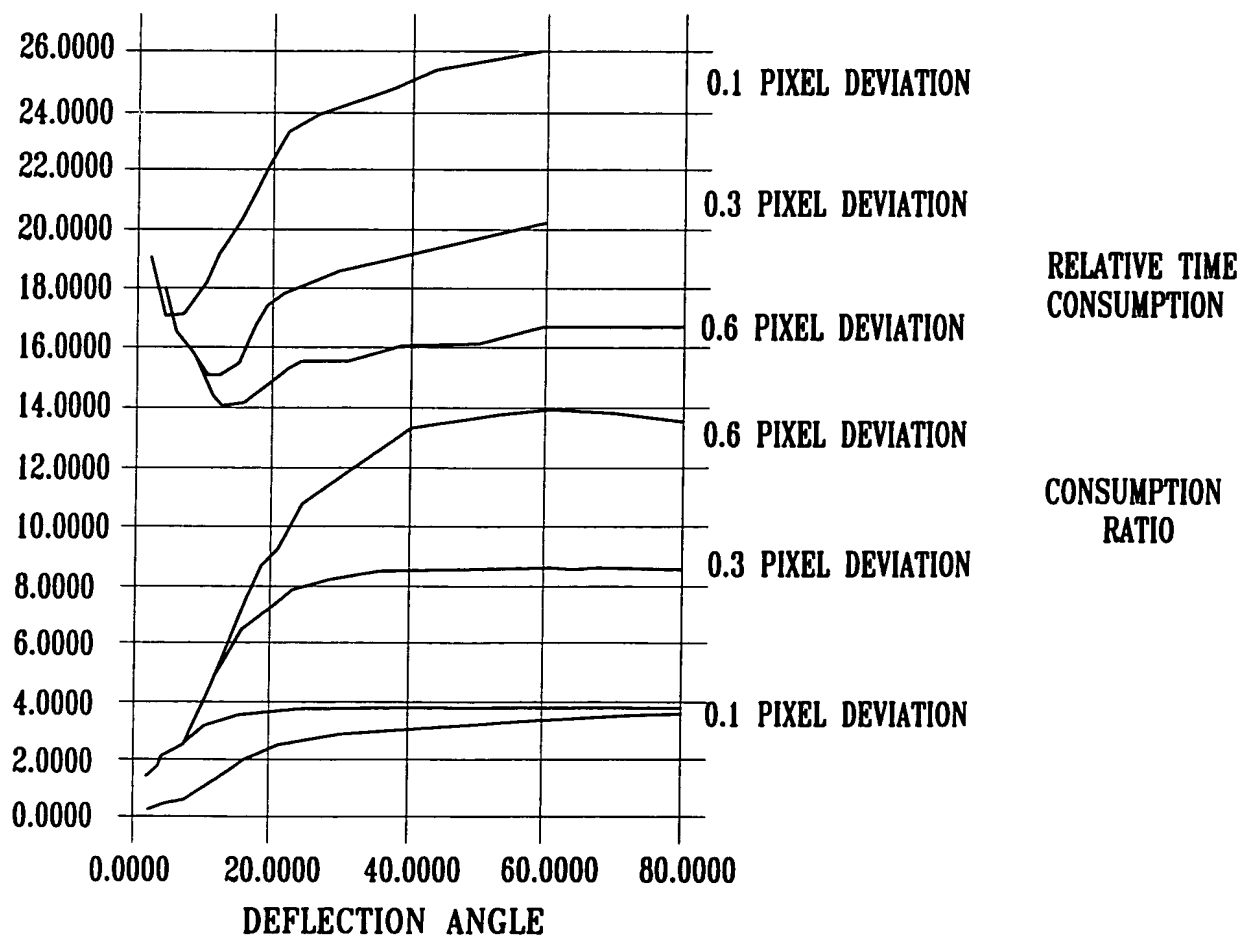


FIG. 81

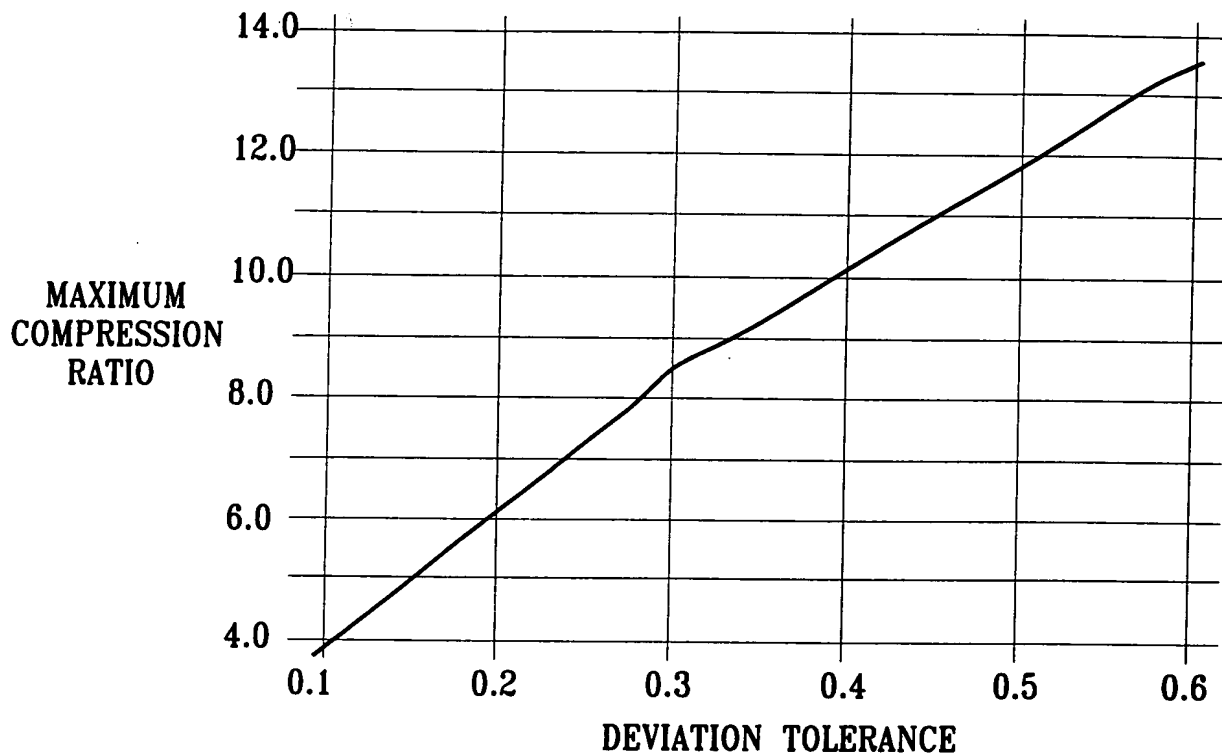


FIG. 82

